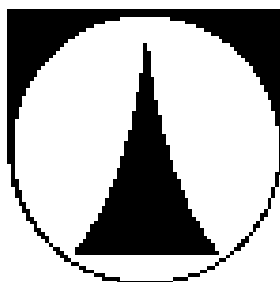


TECHNICKÁ UNIVERZITA V LIBERCI
FAKULTA MECHATRONIKY A MEZIOBOROVÝCH INŽENÝRSKÝCH
STUDIÍ



DIPLOMOVÁ PRÁCE

**Návrh hardware a realizace software řídicí jednotky
systému MDCS**

Igor Pokorný

Liberec

leden 2003

Anotace:

Téma: Návrh hardware a realizace software řídicí jednotky systému MDCS

Tato práce se zabývá konstrukcí hlavní ústředny systému MDCS. Systém MDCS se používá v nemocnicích a domech s pečovatelskou službou a slouží ke komunikaci pacientů s personálem.

V teoretické části jsou rozebrány vlastnosti systému MDCS , jeho protokolu MEDIBUS, popis sběrnice RS 485 a standardní AT klávesnice. Dále jsou popsány součástky, ze kterých je ústředna zkonstruována. Velká část je věnována mikroprocesoru ATMEL 89C55.

Praktická část seznamuje s hardwarovým zapojením hlavní ústředny a popisem programu. Program je napsán v assembleru 8051 a jsou zde popsány některé významné rutiny.

Annotation:

Theme: Design of hardware and software implementation of control unit of MDCS system

This work deals with construction of main exchange of MDCS system. This system is in use in health institutions, like hospitals, old folks houses etc. It provides communication between personnel and patients.

In the theoretical part are analysed attributes of MDCS system, its protocol MEDIBUS, description of RS 485 and standard AT keyboard. Also describes parts, from which is main exchange constructed. A large part is about ATMEL 89C55 microcontroller.

The experimental part presents hardware wiring of the main exchange, also describes the software – only the important routines. Program was written in 8051 assembler language.

MÍSTOPŘÍSEŽNÉ PROHLÁŠENÍ

"Místopřísežně prohlašuji, že diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury".

V Liberci

.....

Igor Pokorný

Poděkování

Na tomto místě bych rád vyjádřil své díky panu ing.Jaroslavu Buchtovi za jeho pomoc a cenné připomínky v průběhu mé práce. Nemohu opominout ani poděkování panu ing.Zátopkovi za poskytnutí potřebných informací.

Použité zkratky

obr.	obrázek
Tab.	tabulka
tzn.	to znamená
Pozn.	poznámka
viz	odkaz na ..
tj.	to je
HÚ	hlavní ústředna
SIJ	signalizační jednotka

Obsah

1.ÚVOD.....	8
2.TEORETICKÁ ČÁST	8
2.1.Systém MDCS	8
2.1.1.Popis signálové jednotky	8
2.1.2.Volací signály.....	10
2.1.3.Funkce hlavní ústředny	10
2.1.3.1.Servisní nastavení	12
2.2.Všeobecný popis a základní parametry RS485.....	12
2.3.Popis protokolu MEDIBUS	13
2.3.1. Interrupt impuls,podmínky kdy ke komunikaci dochází	16
2.3.2. Načtení adresy periferní ústředny	17
2.3.3. Časování sběrnice	18
2.3.4. Další aspekty a připomínky.....	19
2.3.5. Ošetření chyb při přenosu	20
2.3.6. Seznam používaných funkcí a typ předávaných dat	20
2.4.Popis AT klávesnice.....	21
2.4.1. Kódování kláves podle sady 2.....	25
2.5. Návrh hardware.....	27
2.5.1.Popis procesoru ATMEL 89C55	27
2.5.1.1.Organizace paměti.....	28
2.5.1.2.Registry speciálních funkcí.....	29
2.5.1.3.Čítače,časovače.....	29
2.5.1.4.Přerušení	30
2.5.1.5.Sériový kanál.....	31
2.5.1.6.Struktura a činnost vstupních výstupních bran.....	31
2.5.1.7.Přístup do vnější paměti.....	31
2.5.1.8.Instrukční sada.....	32
2.5.2.Popis displeje.....	36
2.5.3.Paměť EEPROM.....	37
2.5.4.Externí RAM,linkový budič,stabilizátor napětí.....	39
3.PRAKTICKÁ ČÁST	40
3.1.Zapojení hlavní ústředny	40
3.2.Popis programu	42
3.2.1.Popis rutin na práci s EEPROM	42
3.2.2.Popis rutin na LCD.....	42
3.2.3.Inicializační část.....	43
3.2.4.Přerušovací rutina kláves od vnějšího přerušení 1	44
3.2.5.Část servisního zadávání.....	45
3.3.Komunikační část programu.....	50
3.3.1.Používané rutiny pro komunikaci.....	50
3.3.2.Popis komunikační části programu.....	53
3.3.3.Rutina KOMUNIK.....	53
3.3.4.Rutina WYZNAC.....	57
3.3.5.Funkce tlačítka na HÚ.....	58
3.3.6.Ošetření chyb.....	59
4.ZÁVĚR	60
5.SEZNAM LITERATURY.....	61

1.Úvod

V provozu nemocnic, hospiců či domovů důchodců se běžně setkáváme s různými komunikačními prostředky, které slouží pacientům k přivolání personálu.

V rámci modernizace těchto zařízení uživatele požadují nové a komplexnější produkty, které nahradí stávající. Uživatelé požadují spolehlivost, nenáročný servis a bezproblémovou montáž. Personál potřebuje mít přehled, který pacient konkrétně žádá obsluhu, případnou možnost dorozumět se hlasovým interkomem, k tomu zvukovou i světelnou signalizaci. Pacienti žádají jednoduché ovládání, uvítají možnost telefonovat do veřejné telefonní sítě, nebo poslouchat zábavný program z jejich komunikačních jednotek. Toto jsou poslední trendy, vystavované na specializovaných veletrzích (Pragomedica) produktů pro zdravotnické zařízení.

2.Teoretická část

2.1.Systém MDCS

Příkladem takového systému může být například MDCS, jehož vývojem a výrobou se zabývá firma ZPT Vigantice. Tento systém umožňuje akusticko-optickou signalizaci volání pacienta. Na centrálním pracovišti, což může být např. sesterna, se umísťuje hlavní ústředna (dále jen HÚ). Na jednotlivých pokojích pacientů jsou umístěny u vstupních dveří signalizační jednotky (dále jen SIJ). Datová komunikace mezi HÚ a SIJ je realizována sběrníci RS485. Používá přitom přenosový protokol označený jako MEDIBUS2.

2.1.1.Popis signalizační jednotky

SIJ plní následující funkce:

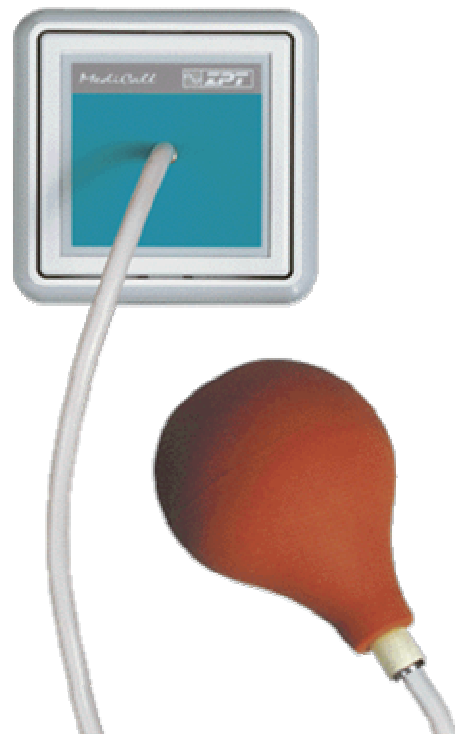
1) předává informace o patientském volání z daného pokoje na HÚ. Volání se aktivuje stiskem tlačítka se symbolem sestry na SIJ. K SIJ lze připojit i samostatná volací tlačítka (paralelně k tlačítku na SIJ). Těmito prvky může být např. volací šňůra (obrázek č.1), tlačítko nouzového volání, pneumatický spínač (na obrázku č. 2).

Samotná SIJ je vyobrazená na obr 3.

Obr 1. : Volací šňůra



Obr 2. : Pneumatický spínač



2) umožňuje registraci sestry. Při příchodu sestry na pokoj se zaregistruje stiskem tlačítka přítomnost (se symbolem tužky) – rozsvítí se LED u tlačítka. Při odchodu se stiskem téhož tlačítka odregistruje – LED zhasne. Registrace sestry na pokoji má několik významů:

a) je-li na daném pokoji aktivováno patientské volání, pak v okamžiku zaregistrování sestry je toto volání zrušeno.

b) informace o registraci sestry se předává na HÚ, kde se vypisuje na LCD.

c) na SIJ, kde je registrace sestry, je akusticky signalizováno volání z jiných pokojů.

d) je-li sestra zaregistrována, lze z dané SIJ vyslat signál alarmu stiskem tlačítka alarm na SIJ (označeno symbolem zvonku). Signál alarmu se zruší opětovným stiskem tlačítka alarm, nebo odregistrací sestry.

3) Je-li na dané SIJ aktivováno nouzové volání nebo alarm, vydává SIJ akustický signál (sama o sobě, tj. aniž HÚ pošle příkaz). SIJ dále může vydávat tón, jehož typ mu pošle HÚ v bytu TN (viz část Popis protokolu MEDIBUS2). Pro následné zrušení tohoto tónu mu HÚ musí vyslat příkaz.

4) Každá SIJ má svoji jedinečnou hardwarovou adresu v rozsahu 1-255. Ta je vždy uvedena na spodní straně SIJ [1].

Obr 3. : Signalizační jednotka



2.1.2. Volací signály

Jak vyplývá z výše uvedeného, v systému mohou existovat dva typy volacích signálů – nouzové patientské volání a alarm. Alarm slouží výhradně pro potřeby personálu. Oba volací signály jsou od sebe odlišeny akustickým tónem. Alarm má vyšší prioritu, tzn. pokud bude v systému aktivován současně alarm a nouzové patientské volání, bude na HÚ aktivní akustický signál alarmu [1].

2.1.3. Funkce hlavní ústředny

HÚ je osazena LCD displejem 128x64 bodů. V uživatelském režimu jsou tyto základní zobrazovací módy:

a) V klidovém stavu (tzn. není žádné volání v systému ani registrace sestry na pokoji) se vyobrazuje „uvítací obrazovka“ s přáním příjemné služby.

b) Je-li na jednom či více pokojích registrace sestry, zobrazují se na LCD údaje o registracích a to vždy číslo místnosti a název místnosti.

c) Je-li v systému nouzové patientské volání či alarm zobrazí se údaje o volání (číslo místnosti a název, typ volání). V případě více volání se řadí volání na displeji v pořadí jak byla aktivována, ale alarm se zařadí vždy před patientské volání, i když byl aktivován později.

HÚ obsahuje tlačítko pro rušení volání, kterým lze zrušit volání, které je první v řadě (ať již nouzové patientské volání, či alarm) a další tlačítka pro pozdější rozšíření systému (např. pro volbu sdruženého provozu).

HÚ dále obsahuje generátor tónu, který vydává akustický signál v případě volání.

HÚ zabezpečuje také aktivaci akustického signálu na SIJ, kde je zaregistrována přítomnost sestry, pokud existuje v systému volání. Následně pak zabezpečuje rušení těchto signálů na SIJ.

Ilustrační příklad sekvence událostí v systému

1) Na pokoji č. 25 se zaregistruje sestra.

- SIJ 25 pošle na HÚ žádost o načtení dat. HÚ načte data z této SIJ, zjistí, že došlo k registraci a zapíše údaj o registraci na displej

2) Pacient z pokoje 33 stiskne tlačítko nouzového volání

- SIJ 33 pošle na HÚ žádost o načtení dat. HÚ načte data ze SIJ33, zjistí, že volá pacient, vypíše informaci o volání na displeji, spustí generátor akustického signálu (tón nouzového volání) a dále pošle na SIJ 25, kde je zaregistrovaná sestra, příkaz ke spuštění akustického signálu nouzového volání.

Pozn. 1: Na SIJ 33 HÚ příkaz ke spuštění akustického tónu nevysílá, SIJ 33 vydává akustický tón sama.

Pozn. 2: Pokud by bylo více SIJ v systému, kde je registrace sestry, je potřeba vyslat příkaz k tónovému návěští na všechny (pokud tak nebylo učiněno dříve)

3) Na pokoj 33 přijde sestra a zaregistruje se.

- SIJ 33 pošle na HÚ žádost o načtení dat. HÚ načte data ze SIJ 33, zjistí, že byla provedena registrace sestry a že bylo zrušeno nouzové patientské volání. Aktualizuje displej (bude zde informace o registraci sestry na pokojích 25 a 33), vypne generátor tónového návěští a dále vyšle na SIJ 25 příkaz ke zrušení akustického signálu.

Pozn. 3: Pokud HÚ vyslala na SIJ příkaz ke spuštění akustického signálu, musí se postarat i o jeho zrušení. SIJ tedy neruší signál tónového návěští vyslaný HÚ automaticky (ani při odregistrování sestry) .

2.1.3.1.Servisní nastavovací režim

V servisním nastavení se provádí nastavení parametrů systému. Jedná se v první řadě zadání adres SIJ, které jsou obsaženy v systému pro tyto nastavení čísla místnosti a názvu místnosti. Servisní nastavení se provádí přes standardní PC klávesnici, která se k HU připojí. Ke každé SIJ se přiřazuje 4 ciferné číslo pokoje a textový řetězec o délce max. 20 znaků udávající název pokoje. Zadaná data se poté ukládají do EEPROM.

2.2.Všeobecný popis a základní parametry RS 485

Základní parametry rozhraní RS 485 jsou definovány normou EIA. Podle normy umožňuje linka RS 485 komunikaci mezi 32 účastníky na celkovou vzdálenost do 1,2 km. Pro více účastníků je nutno použít opakovací. Linka je symetrická, diferenciallyně buzená – informace je přenášena po dvou vodičích (označených jako A a B), přičemž signály jsou vysílány v protifázi. K přenosu se používají twistované (tj. kroucené) vodiče. Vlastní komunikaci zajišťují obvykle integrované obvody umožňující převod TTL signálů na úroveň RS 485, které sdružují jak vysílače, tak přijímače, např. ADM 485. Tento obvod má následující vývody :

A – neinverující vstup přijímače / výstup vysílače

B – invertující vstup přijímače / výstup vysílače

RO – vstup z přijímače

/RE – log.1 na tomto vstupu uvede výstup přijímače do stavu vysoké impedance

DI – vstup vysílače

DE – je-li tento vstup v log. 0, je vysílač ve stavu vysoké impedance. Je-li zapojeno více obvodů na tuto linku, může být v daném okamžiku vybaven pouze jeden vysílač, ostatní musí být ve stavu vysoké impedance

Vcc – kladný pól napájení (+5V)

GND – záporný pól napájení

Symetrická linka má tu výhodu, že je odolná proti souhlasnému rušivému napětí indukovanému do linky a také vůči rozdílnému potenciálu země vysílače a země přijímače. Logická úroveň na výstupu přijímače je dána pouze rozdílem napětí mezi linkami A a B, přičemž prahové napětí je okolo 0V. Na výstupu přijímače RO bude logická 0, bude-li na lince A napětí nižší minimálně o 200 mV než na lince B. Na výstupu RO bude logická 1, bude-li napětí na lince A minimálně o 200 mV vyšší než na lince B. Oblast napětí v rozmezí od -200 mV do 200 mV mezi linkami představuje neurčitou oblast.

Podle normy musí být schopen vysílač dodat na linky A a B diferenční napětí minimálně 1,5V a to i v případě, že je linka zakončena na obou koncích rezistorem 120 ohm [2].

2.3.Popis protokolu MEDIBUS2

Komunikace je simplexní, tzn. že se využívá pouze jeden komunikační kanál (jeden twist) po kterém se předávají data obousměrně. Hlavní ústředna a každá periferní ústředna připojená na systémovou sběrnici je vybavena rozhraním RS485 obsahující jeden vysílač a jeden přijímač. V klidovém stavu jsou všechny vysílače ve stavu vysoké impedance. Pro správnou funkci je však potřeba, aby v klidovém stavu byly na výstupech přijímačů úrovně log. 1. To je zajištěno prostřednictvím pull-up a pull-down rezistorů na lince A a B (tyto se umísťují spolu s rezistorem zakončující vedení zpravidla v poslední periferní ústředně na vedení). Rezistory vnutí linkám takové napětí, aby linka A měla minimálně o 200 mV vyšší napětí, než linka B. Tento stav interpretují přijímače jako log. 1. Velikost pull-up a pull-down rezistorů je dostatečně vysoká na to, aby v případě, že je vybaven některý z vysílačů, neměly

rezistory podstatný vliv na správné úrovně na linkách A a B (nebo-li vnitřní odpor vysílačů je mnohem menší než hodnota pull-up a pull-down rezistorů).

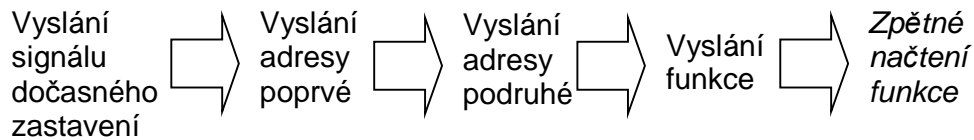
Komunikaci na sběrnici řídí vždy HÚ, tzn. že jednak komunikaci začíná a jednak kontroluje korektní průběh komunikace. V případě, že detekuje chybu při komunikaci, sama vyvolává opravné mechanismy. Nicméně periferní ústředny (SIJ) mají možnost vyslat žádost o komunikaci (prostřednictvím interrupt impulsu), pokud jsou splněny definované podmínky na sběrnici. Veškeré informace se přenáší po bytech. Formát přenosu jednoho byte je následující :

startbit -> 0.bit -> 1.bit -> 2.bit->...->8.bit -> stopbit

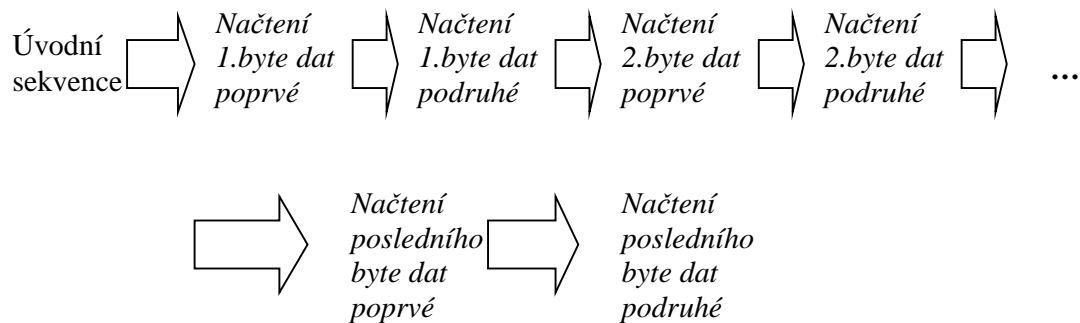
Osmý bit se nastavuje do log. 1 u adresy, a do log. 0 u dat a funkce. Periferní ústředny kontrolují u přijatého byte, zda je 8. bit správně nastaven.

HÚ začíná komunikaci tím, že vyše po lince signál dočasného zastavení, tj uvedení linky do log. 0 na dobu t_2 . Tímto dá periferním ústřednám signál, že bude vysílat adresu. V dalším kroku vyše HÚ adresu periferní ústředny se kterou miní komunikovat. Adresa se vysílá dvakrát po sobě pro vyloučení chyby. Pokud periferní ústředna zjistí u obou přijatých byte shodu se svou nastavenou adresou, připraví se na další komunikaci. Dále vyše hlavní ústředna byte FUNKCE, který určuje objem a typ přenášených dat v následujících krocích. Nyní zaadresovaná periferní ústředna vyše pro kontrolu přijatou funkci zpět na HÚ. HÚ zkontroluje, zda je v přijatá funkce stejná jako vyslaná. Pokud není, nebo pokud do doby t_5 není přijatý žádný byte, pokouší se HÚ navázat spojení s periferní ústřednou znovu dle výše popsaného schématu. Bit D7 funkce rozlišuje, zda se jedná o funkci načítání dat z periferní ústředny ($D7 = \text{log}.0$) či funkci zápisu dat na periferní ústřednu ($D7 = \text{log}.1$). Typ funkce dále určuje, zda je přenášen vždy stejný počet datových byte, nebo zda je počet přenášených datových byte variabilní. Druhý případ však v této práci není užít, proto dál již zmíněn nebude. Následuje přenos datových byte. U funkcí, které provádějí pouze čtení byte z periferní ústředny, je každý byte načten pro kontrolu dvakrát. V případě, že nejsou načtené byte shodné, provádí se čtení z dané periferní ústředny znovu (opakuje se celá přenosová sekvence, počínaje signálem dočasného zastavení) [2].

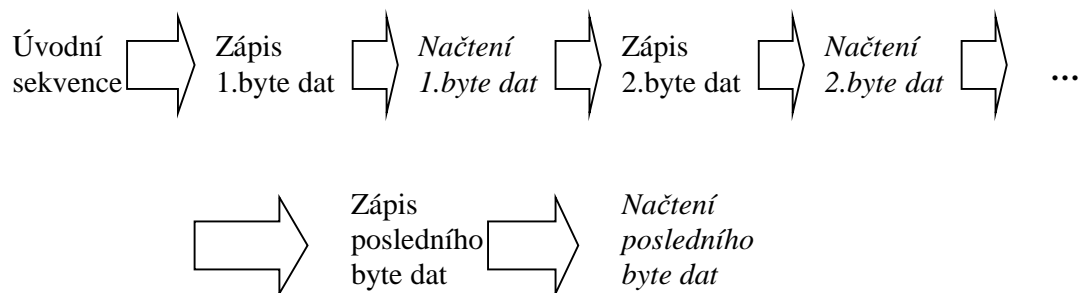
Úvodní sekvence je vždy stejná (výjimku tvoří načítání adresy periferní ústředny – viz níže) a probíhá takto:



Pro funkce čtení s pevným počtem přenášených dat je sekvence následující:



Pro funkce zápisu s pevným počtem přenášených dat je sekvence následující:



Pozn.: U popisů psaných *kursivou* jsou data vysílána periferní ústřednou. U ostatních jsou data vysílána HÚ.

2.3.1. Interrupt impuls, podmínky za kterých ke komunikaci dochází

Doposud byl popsán pouze mechanismus komunikace mezi hlavní ústřednou a vybranou periferní ústřednou, což představuje základní prvek komunikace. Nebylo však probráno, kdy ke komunikaci dochází. Ke komunikaci dochází v těchto dvou případech :

1) V okamžiku, kdy dojde ke změně stavu libovolné periferní ústředny – HÚ musí tuto změnu zaregistrovat. Změnou stavu se rozumí např. zaregistrování se sestry, odregistrování se, spuštění nouzového volání, atd. Při změně stavu musí tedy HÚ načíst nový aktuální stav periferní ústředny. Jak již bylo uvedeno, v klidovém stavu není na linku RS485 připojen žádný vysílač, pouze přes rezistory je zajištěn definovaný stav linky – log.1. V okamžiku, kdy dojde ke změně stavu libovolné periferní ústředny, testuje daná periferní ústředna po dobu t_7 stav linky. Pokud je po danou dobu linka trvale ve stavu log.1, znamená to, že neprobíhá zrovna žádná komunikace – linka je tedy v klidovém stavu (není-li linka v klidovém stavu, musí periferní ústředna vyčkat, až tento stav nastane). V tomto okamžiku připojí periferní ústředna svůj vysílač na linku a vyšle log. 0 o definované šířce. Dále bude tento impuls označován jako interrupt impuls. Hlavní ústředna tento impuls zachytí a zjistí adresu periferní ústředny, která žádost vyslala. V případě, že nebudou nová data z periferní ústředny načtena v požadované době t_8 , pokouší se periferní ústředna vyslat interrupt impuls znovu (může dojít k situaci, kdy interrupt impuls zasáhne do signálu dočasného zastavení a není tedy zaregistrován). Po načtení stavu periferní ústředny je provedeno vyhodnocení a podniknuty příslušné kroky. Interrupt impuls, přestože jeho vysílání není synchronizováno s ostatními prvky, nemůže způsobit ani v nejnepříznivějším případě, tj. kdyby současně začala vysílat HÚ, kolizi na sběrnici. Je to dáno za prvé tím, že periferní ústředna monitoruje po dobu t_7 stav sběrnice, zda je skutečně v klidu (z toho vyplývá důležitá podmínka, kterou je nutno dodržet při datové komunikaci a to, že při komunikační sekvenci nesmí přesahovat doba vysílání dvou po sobě následujících byte stanovenou mez). Za druhé, i kdyby současně začala hlavní ústředna komunikační sekvenci, překryje signál dočasného zastavení vysílaný z hlavní ústředny interrupt impuls.

2) Když je nutno na základě situace vyslat na danou periferní ústřednu data modifikující její výstupy. Tj. na některé periferní ústředně dojde ke změně stavu,

periferní ústředna vyšle interrupt impuls, HÚ načte data, tyto zpracuje a může dle aktuální situace vyslat na některé periferní ústředny aktualizovaná data [2].

2.3.2.Načtení adresy periferní ústředny

K načtení adresy periferní ústředny, která interrupt impuls vyslala, se provádí následujícím způsobem. HÚ zahájí vysílání úvodní sekvence tak, jak byla popsána výše s tím, že hodnota adresy i funkce je 0. Úvodní sekvence je stejná až do okamžiku vyslání funkce, poté očekává hlavní ústředna příjem dvou byte v kterých je udána informace o adrese periferní ústředny, která žádá o obsluhu. Pokud periferní ústředna žádá o obsluhu, tak po stanovené době po přijetí funkce zahájí vysílací sekvenci. Vysílací sekvence je ze strany periferní ústředny nestandardní (ze strany HÚ se však jeví jako standardní) a to z toho důvodu, že vysílání může zahájit najednou více periferních ústřed, které žádají o obsluhu. I v tomto případě je zajištěno bezkolizní odeslání informace o adrese, přičemž se prosadí periferní ústředna s nejnižší adresou. Vysílací sekvence začíná vysláním start bitu. Ten vysílají všechny periferní ústředny žádající o obsluhu. Aby byla dodržena synchronizace, je nutno dodržet přesně stanovený čas vyslání start bitu od stop bitu přijaté funkce. V bitu D0 je vysílána ze všech zúčastněných periferních ústřed úroveň log. 1. V bitu D1 je vysílán nejvyšší bit adresy, tj. A7, periferní ústředny. Vysílání se zúčastní pouze periferní ústředny u kterých je A7 log. 0. Ostatní, u kterých je A7 log. 1 odpojí svůj budič linky a sledují stav na lince. Pokud zjistí, že v bitu D1 je vysílána log. 0, je přenosová sekvence ukončena, pokud je v bitu D1 načtena log. 1, pokračují v přenosové sekvenci počínaje následujícím bitem. V bitu D2 vysílají všechny zúčastněné periferní ústředny, které neukončily vysílání, log. 1. V bitu D3 se vysílá bit A6 adresy periferní ústředny, přičemž platí stejná pravidla jako při vysílání bitu A7 adresy. Analogicky se pokračuje až do vyslání bitu A4 adresy, za kterým je vyslán stop bit. Dále následuje vyslání dvou bitů s logickou úrovní 1. Z pohledu HÚ je tedy v tomto okamžiku vyslán jeden standardní byte, i když se vysílání účastní více periferních ústřed. Následně je vyslán obdobným způsobem druhý byte, který nese informaci o nižších čtyřech bitech adresy.

Pokud žádá více periferních ústřed o obsluhu, zpřístupní se adresa další periferní ústředny až po obsluhu periferní ústředny s nejnižší adresou. HÚ musí mít nastaven time-out, aby došlo k ukončení přenosu i v případě, že žádná periferní ústředna o obsluhu nežádá [2].

2.3.3.Časování sběrnice

Vzhledem k tomu, že ve výše uvedeném režimu může vysílat současně více periferních ústředen, je nutno přesně dodržet časování, aby bylo zajištěno synchronní vysílání [2].

Tab.1 : Časování sběrnice

Symbol	Hodnota	Popis
t1	52 us	Doba přenosu jednoho bitu (přenosová rychlost 19200 baudů).
t2	1700 us	Doba trvání signálu dočasného zastavení vysílaného HÚ.
t3	575 us	Minimální doba trvání log. 0 na sběrnici, aby periferní ústředna akceptovala tento signál jako signál dočasného zastavení .
t4	100 us	Minimální doba, po které musí zůstat linka v klidu mezi dvěmi po sobě vysílanými byte z HÚ, popřípadě mezi přijetím byte a vysláním byte (čas mezi koncem stop bitu jednoho byte a začátkem start bitu následujícího).
t5	1280 us	Maximální doba, po kterou může v rámci komunikační sekvence zůstat linka v klidu mezi dvěmi dvěma po sobě vysílanými byte, popřípadě mezi přijetím byte a vysláním byte (čas mezi koncem stop bitu jednoho byte a začátkem start bitu následujícího).
t6	216 us	Šířka interrupt impulsu vysílaného periferní ústřednou.
t7	1280 us	Minimální doba, po kterou musí být linka v klidu, aby mohla vyslat periferní ústředna interrupt signál.
t8	0,5 s	Doba, za kterou se musí pokoušet vyslat periferní ústředna opět interrupt signál, pokud nebyl akceptován interrupt signál vyslaný v předchozím kroku.
t9	1280 us	Minimální doba, po kterou musí být linka v klidu, aby mohla hlavní ústředna opakovat přenos po detekci chyby přenosu.
t10	156 us	Vzdálenost začátku stop bitu – začátek start bitu při vysílání adresy periferní ústředny.

2.3.4. Další aspekty a připomínky

1) Signál dočasného zastavení

U většiny periferních ústředen je sériové rozhraní implementováno softwarově, což vyžaduje pravidelné testování linky. Z tohoto důvodu předchází každé komunikační sekvenci s danou periferní ústřednou signál dočasného zastavení. Ten dává informaci, že bude probíhat komunikace. Hlavní ústředna vysílá signál dočasného zastavení o délce t_2 . Periferní ústředna však akceptuje signál dočasného zastavení pokud má délku alespoň t_3 . Délka t_3 je vyšší než odpovídá přenosu jednoho byte, takže lze při testování linky na přítomnost signálu dočasného zastavení jednoznačně určit, zda se jedná skutečně o signál dočasného zastavení, nebo jde pouze o komunikační sekvenci mezi HÚ a některou periferní ústřednou. Vzhledem k tomu, že t_3 je menší než t_2 , není potřeba, aby periferní ústředna zachytila signál dočasného zastavení ihned. Toto dává dostatečný prostor pro vykonávání dalších podprogramů na periferní ústředně. Testování sběrnice na přítomnost signálu dočasného zastavení musí periferní ústředna začít provádět nejpozději za dobu $t_2 - t_3$ od posledního testování (samozřejmě pokud právě neprobíhá komunikační sekvence s touto periferní ústřednou). Jedině tak nemůže dojít k přehlédnutí signálu dočasného zastavení ze strany periferní ústředny.

Periferní ústředna musí být schopna rozpoznat signál dočasného zastavení i pokud se vyskytne v průběhu komunikační sekvence mezi HÚ a ní. V tomto případě začíná celou přenosovou sekvenci znovu, bez ohledu na to, zda byla předchozí přenosová sekvence ukončena.

2) Time – outy

Pokud dojde během přenosu k překročení některého z definovaných časů, je přenos ukončen. HÚ se při time-outu snaží navázat komunikaci opět, stejně jako při výskytu chyby [2].

2.3.5.Ošetření chyb při přenosu

Při načítání dat z periferní ústředny se čte každý byte pro kontrolu dvakrát. Při zápisu byte na periferní ústřednu se ihned po zápisu vyšle zapsaný byte zpět na HÚ. Pokud detekuje HÚ chybu, ukončí přenos a opakuje celou komunikační sekvenci s danou periferní ústřednou znovu.

Chybu rámce (např. neplatný stop bit, neplatný 8. bit) je schopna rozpoznat i periferní ústředna. Pokud zjistí periferní ústředna, že došlo k chybě, přestane komunikovat. HÚ, která čeká na data z periferní ústředny, zjistí po uplynutí timeoutu, že periferní ústředna neodpovídá, a zopakuje celou přenosovou sekvenci znovu.

Celkem se učiní 5 pokusů. Před opakováním přenosové sekvence musí hlavní ústředna vyčkat do okamžiku, kdy je linka v klidu po dobu min. t_9 (čeká se z toho důvodu, že periferní ústředna může ještě vysílat další byte dokončující přenosovou sekvenci) [2].

2.3.6.Seznam používaných funkcí a typů předávaných dat

Komunikační protokol MEDIBUS2 počítá s řadou periferních ústředen. Tato práce však měla za úkol pouze vyřešit komunikaci mezi HÚ a SIJ. Z tohoto důvodu bude seznam funkcí a typy předávaných dat omezen výlučně na tento případ [2].

Seznam funkcí

Funkce:	Význam:
11H	čtení proměnné BKOTL
88H	zápis se zpětným čtením proměnné TN
A0H	resetování periferní ústředny – z hlediska sekvence předávaných dat se tato funkce považuje za standardní funkci zápisu jednoho byte dat, přičemž jako data se předává hodnota A0H. periferní ústředna provede reset v případě, že i přijatý byte dat je korektní.

Byte TN:

Předává se zde kód tónového návěští

- bity D0, D1, D2, určují typ tónového návěští:

D2	D1	D0		
0	0	0	-	žádné tónové návěští
0	1	0	-	tónové návěští nouzového volání
0	1	1	-	tónové návěští alarmu

Ostatní bity mají jinou funkci, která není v SIJ implementována (např. hlasitost)

Načítané byte z periferní ústředny**Byte BKOTL:**

Bit D0 – udává stav bistabilního klopného obvodu tlačítka SLUŽEBNÍ ALARM

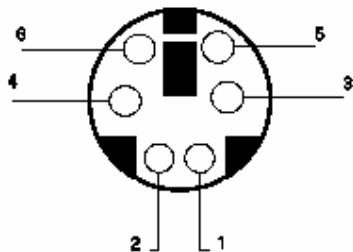
Bit D2 – udává stav bistabilního klopného obvodu tlačítka (mg. spínače)

PŘÍTOMNOST

Bit D4 - udává stav bistabilního klopného obvodu tlačítka NOUZOVÉ VOLÁNÍ

2.4. Popis AT klávesnice

K přiřazení názvu a čísla pokoje k jednotlivým SIJ byla v zadání diplomové práce požadována možnost připojit k HÚ standardní AT klávesnici. Samotné připojení je realizováno konektorem minidin (PS/2). Tento konektor má 6 vývodů, používají se však jenom 4. Viz obrázek č.4. Vývod číslo 2. je používán k obousměrnému přenosu sériových dat mezi klávesnicí a nadřazeným obvodem. Hodinová frekvence synchronizující tento přenos dat je generována mikrořadičem klávesnice a posílána na vývod č 5. Dalšími nezbytnými vývody jsou č. 4 a 5. Číslo 4. je zemnění, na 5. se přivádí napětí 5V pro napájení klávesnice [3].



Obr. č. 4: Popis konektoru minidin (PS/2), samice

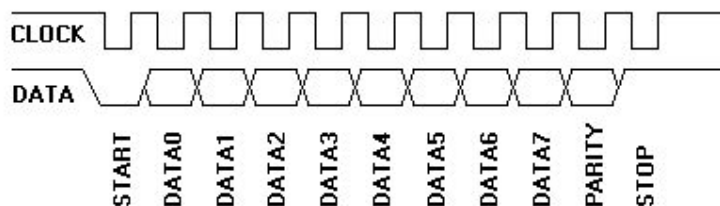
- 1. DATA
- 3. Země
- 4. +5V
- 5. CLK
- 2,6. Neimplementováno

Klávesnice se skládá z velké matice kláves, která je monitorována klávesnicovým mikrořadičem. Mikrořadič klávesnice se je různý podle typu a výrobce klávesnice. Jsou to například mikrořadiče INTEL: 8048,8049, nebo MOTOROLA: 6868, 68HC11, 6805, či HOLTEK: HT82K28A, HT82K628A. Všechny však vykonávají stejnou funkci. Neustále sledují stav matice spínačů. V případě jakékoliv změny (zmáčknutí, puštění, nebo držení) zjistí mikrořadič souřadnice spínače, který ji způsobil, a přiřadí příslušný SCAN-kód. Každá klávesa má své dva unikátní kódy a to při zmáčknutí/držení, nebo puštění. Ten se po sériové lince DATA vyše směrem k systémové desce.

Existují celkem tři SCAN-kódovací sady. První používaly dnes již zastaralé XT (IBM XT keyboard 1981) klávesnice, druhá sada je nejpoužívanější a nejrozšířenější (IBM AT keyboard 1984), třetí je málo používaná PS/2 sada (IBM PS/2 keyboard 1987). Většina klávesnic používá druhou sadu. Z hlediska této práce je zajímavá pouze druhá sada, proto jsou její SCAN-kódy uvedeny v příloze.

Data jsou synchronizována hodinovým signálem CLK, který má hodnotu od 10 až 33 kHz, záleží opět na typu klávesnice. Komunikuje se po 11 bitech. První se vysílá startbit (logická 0), následuje 8 datových bitů (první jde LSB), paritní bit (lichá parita) a končí se stopbitem (logická 1). Viz obrázek č. 5.

Obr. č. 5: Synchronizace dat vysílaných klávesnicí



Tato synchronní sériová komunikace je obousměrná, což znamená, že podobným způsobem se mohou vysílat data do klávesnice. Je třeba podržet signál DATA na log. 0 do té doby, než generátor hodin klávesnice začne vysílat své pulsy. Pak stačí sledovat tento signál a posílat bity.

Jestliže zmáčkne nějakou klávesu, mikrořadič vyšle její SCAN-kód. Pokud bude stále zmáčknuta, bude mikrořadič vysílat stále její kód, dokud nebude puštěna, nebo nebude zmáčknuta jiná klávesa (tzn. z klávesy se z stane *typematická*). Existují tedy dvě prodlevy. První je mezi posláním prvního a opakovaného kódu (v angličtině tzv. *typematic delay*), druhá mezi rychlým vysíláním při delším držení klávesy (*typematic rate*). Prodleva mezi řetězem vysílaných kódů je nastavitelná příkazem poslaných do klávesnice. *Typematic delay* se pohybuje v rozpětí od 0.25 do 1 sekundy, *typematic rate* od 2 do 30 znaků za sekundu [4].

Jak již bylo zmíněno, do klávesnice lze posílat z řídící jednotky různé příkazy. Příkazy jsou jednobytové čísla, je jich 18. Přehled důležitých příkazů a odpovědí je uveden v následujících tabulkách.

Tab. č. 2: významné příkazy pro klávesnici

Kód příkazu v hexadecimálním formátu	Popis příkazu
ED	Zapíná a vypíná LED na klávesnici. Klávesnice odpoví potvrzením příkazu (FA), a čeká na další byte. Tento určuje, stav LED (1-3). Zbytek bytu je doplněn nulami. Pzn: b0- Scrolllock, b1 – NumLock, b2 – CapsLock. Pzn2: log 1 - svítí Log 0 – nesvítí
EE	Echo – klávesnice odpoví rovněž echem (EE)
F0	Nastavení SCAN-kódovacího standardu. Klávesnice odpoví potvrzením (FA) a čeká na další byte, který určuje, kterou sadu kódů má klávesnice používat. Byte nabývá hodnot 01, 02, 03, což odpovídá sadám 1, 2, 3. Jestliže je však místo těchto hodnot v poslána 0, klávesnice odpoví potvrzením a pošle číslo sady, kterou zrovna používá (01,02,03)

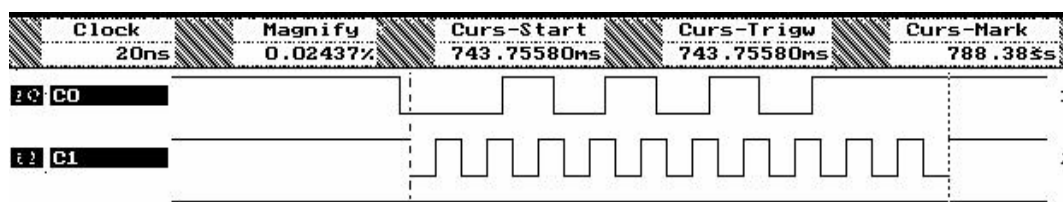
F3	Nastaví opakovací frekvenci – klávesnice odpoví potvrzením,následuje další byte,který určuje, opakovací frekvenci. Bity 0-4 určují opakovací frekvenci (kolikrát za vteřinu se má poslat kód klávesy), bity 5-6 určují délku první prodlevy. 7 bit není implementován. Po vyslání tohoto bytu klávesnice opět odpoví potvrzením.
F4	Povolení klávesnice – vyčistí její zásobník a povolí sledování matice spínačů. Odpoví potvrzením.
F5	Zakázání klávesnice – zakáže sledovat matici spínačů.Odpoví potvrzením.
F6	Obnovení implicitních hodnot – obnoví se původní nastavení klávesnice. Netýká se LED.
F7	Nastaví všech kláves na <i>typematické</i> . Klávesnice odpoví potvrzením.
F8	Zakázání <i>typematickosti</i> kláves. Při zmáčknutí se vysílá pouze kód zmáčknutí a kód puštění. Klávesnice odpoví potvrzením.
F9	Po tomto příkazu se při mačkání kláves bude vysílat pouze jejich kód zmáčknutí. Klávesnice odpoví potvrzením.
FA	Klávesnice počne pracovat tak jak známe, tzn. při rychlém zmáčknutí a puštění kódy klávesy zmáčknutí a puštění, při držení se klávesa stane <i>typematickou</i> . Klávesnice odpoví potvrzením.
FE	Klávesnice pošle poslední odpovídací bajt, který naposledy poslala.
FF	Reset – resetuje klávesnici

Tab. č.3: Přehled některých odpovědí klávesnice

Kód odpovědi v hexadecimálním formátu	Popis
00	Chyba detekce klávesy/přetečení zásobníku. Posílá se při nastavených SCAN-kódovacích sadách 2,3.
AA	Vlastní diagnostický test úspěšně proveden.

EE	Echo – odpověď na Echo z řídicí jednotky
FA	Potvrzení – posílá se za téměř každým obdrženým příkazem
FC	Vlastní diagnostický test neúspěšný.
FE	Přeposlání – řídicí pošle znovu poslední příkaz
FF	Chyba detekce klávesy/přetečení zásobníku. Posílá se při nastavené SCAN-kódovací sadě 1.

Obr. č. 6: vysílání klávesnicové odpovědi po úspěšném diagnostickém testu po zapnutí napájení (AA) nahoře data, dole hodiny.



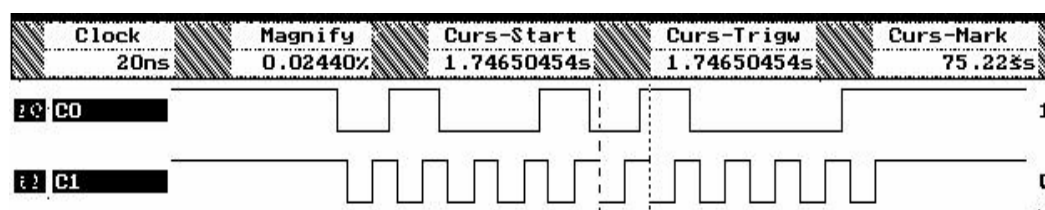
Pro uspokojující řešení úlohy připojit klávesnici k HÚ bylo považováno za vhodné nechat klávesnici pracovat podle takových zvyklostí, které má většina uživatelů. Bylo zachováno implicitní nastavení hodnot prodlev tzn. 0.5 sekundy mezi prvním a druhým vyslání kódu stejné klávesy a frekvenci 10 kódů za sekundu při *typematičnosti*. Dále bylo rozhodnuto používat kódovací sadu 2, tedy tu nejrozšířenější.

2.4.1. Kódování kláves podle sady 2

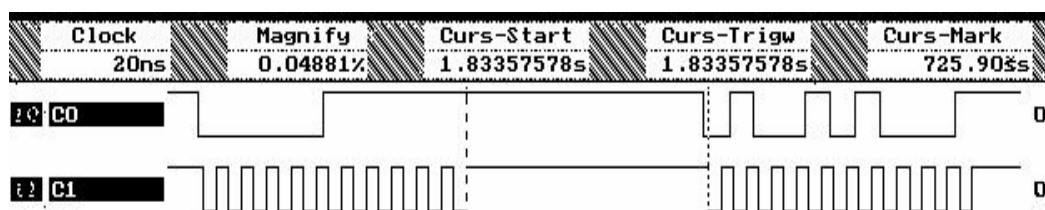
Dnešní klávesnice mívají 101, nebo 102, některé však 104 kláves. Jelikož klávesy pro ovládání PC aplikací pozbývají v této práci smyslu, zabýváme se tedy pouze klávesnicí se 101 klávesami.

Z hlediska kódování kláves klávesnicí je možné klávesy rozdělit do tří skupin. Je to skupina základní, skupina rozšířená a skupina speciální. Do základní skupiny patří 83 kláves, do rozšířené 14 kláves a do speciální 2 (Pause a Printscreen) klávesy. Při stisku a puštění některé ze klávesy ze skupiny základní je vyslán nejprve kód klávesy, poté kód puštění a znovu kód klávesy. Dokumentují to dva následující obrázky z logického analyzátoru LOGAN. Na prvním je zmáčknuta mezera, je tedy vyslán kód mezery (29h). Na druhém obrázku je zachyceno puštění. Nejprve se vyšle kód puštění (F0h), následuje kód mezery.

Obr. č. 7 : Základní klávesa stisknuta

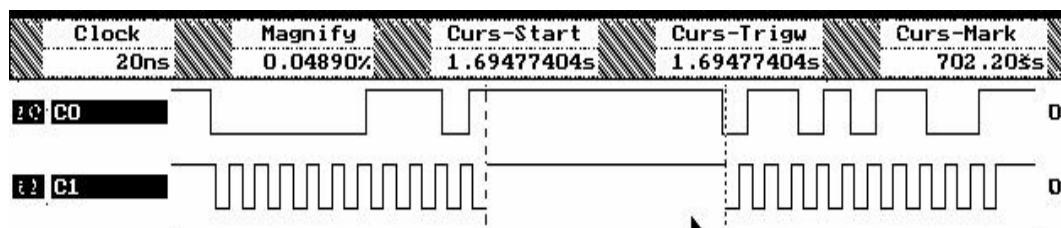


Obrázek č.8 : Základní klávesa puštěna

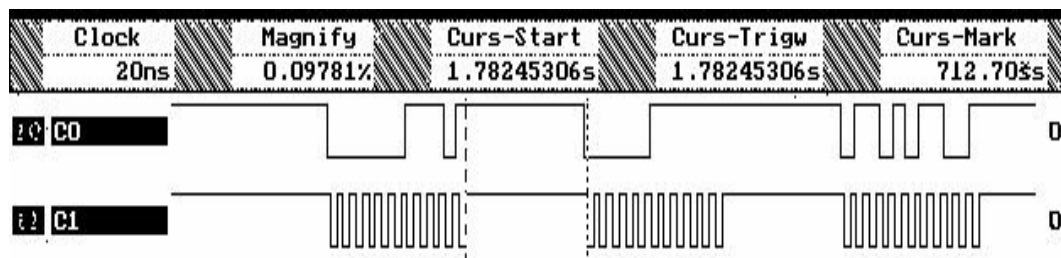


Rozšířená skupina zahrnuje tyto klávesy: Pravý Alt s pravým Ctrl, Insert, Home, Page up & down, Delete, End, Up, Down, Left, Right a Numerický enter a /. Tyto klávesy se vyznačují tím, že sice mají shodné kódy jako klávesy ze základní skupiny, ale po jejich stisku se vysílá nejdříve prefix (E0h). Při puštění se nejdříve vysílá prefix, dále kód puštění a pak teprve samotný kód klávesy.

Obrázek č.9 :Zmáčkнутá klávesa Left (6Bh) s prefixem (E0h)



Obrázek č.10 : Puštění klávesy Left s prefixem (nejdříve prefix, pak kód puštění a kód klávesy).



Klávesy ze speciální skupiny není třeba rozebírat, pro danou úlohu nemají funkci. Jen pro ilustraci, klávesa PrintScreen po zmáčknutí vysílá sekvenci E0h, 12h, E0h, 7Ch, po puštění E0h, F0h, 7Ch, E0h, F0h, 12h [4].

Při současném stisku dvou nebo více kláves (základní a rozšířená skupina) jsou kódy vysílány v odpovídajícím pořadí, v jakém se akce stisku a puštění provedly, a nijak se neovlivňují, kromě případu, kdy je narušeno cyklické vysílání zmáčknuté klávesy (*typematické*). To nastane, je-li klávesa dlouho zmáčknutá a cyklicky se vysílá její kód, až do okamžiku, kdy je zmáčknutá jiná klávesa. Pak se přestane vysílat kód první stisknuté a držené klávesy. Tento případ může nastat například při psaní velkých písmen za pomoci klávesy Shift. Je-li držen déle než 0.5 sekundy, klávesnice vysílá cyklicky jeho kód (12h), následuje stisk a puštění klávesy písmena (např. N – 31h, F0h, 31h), poté puštění klávesy Shift (F0h, 12h).

2.5.Návrh hardware

Při návrhu hardware bylo třeba vybrat odpovídající procesor, který by splňoval nároky na tuto úlohu a cenově odpovídal. Výrobků na našem trhu je mnoho, zvolen byl nakonec procesor firmy ATMEL. Dále bylo nutno zvolit typ EEPROM paměti, displej, linkový budič, případnou externí paměť RAM pro uložení více dat.

2.5.1.Popis procesoru ATMEL 89C55

Srdce HÚ tvoří procesor firmy ATMEL. Jedná se o typ 89C55, která vychází z řady 8052. Jedná se o osmibitový jednočip s harvardským uspořádáním. Datová a programová paměť je oddělená. Procesor je napájen 5V, a jeho takt udává piezokeramický krystal, který se připojuje na vývody XTAL1 a XTAL2. CPU je propojena vnitřní společnou sběrnici s pamětí programu o kapacitě 20K a s 256 byty datové paměti. Styk procesoru s vnějšími periferiemi zajišťují čtyři vstupně/výstupní brány P0 až P3. Procesor má rovněž vývody pro styk s externí datovou (\overline{WR} , \overline{RD}) či programovou pamětí. Řadič přerušení zpracovává pět zdrojů přerušení – dvě externí $\overline{INT0}$, $\overline{INT1}$, od každého časovače, a od sériového kanálu. Jednotlivá přerušení mají možnost definice priorit. Všechny tři čítače jsou šestnáctibitové, jejich hodinový signál se odvozuje od interního generátoru hodin anebo ze vnějších vstupů T0, T1, T2, T2EX. Mají možnost pracovat v několika různých režimech. Pro snazší styk s nadřazeným systémem nebo jinými procesory je

procesor vybaven sériovým obousměrným kanálem. Booleovský procesor umožňuje pracovat s jednotlivými bity vnitřní paměti RAM i interních periférií. Je uzpůsoben pracovat na frekvenci až 24 Mhz.

Obr. č. 11: Schematické znázornění umístění vývodu procesoru 8055

Název vývodu	Číslo	Číslo	Název vývodu
P1.0 T2	1	40	Vcc
P1.1 T2EX	2	39	P0.0
P1.2	3	38	P0.1
P1.3	4	37	P0.2
P1.4	5	36	P0.3
P1.5	6	35	P0.4
P1.6	7	34	P0.5
P1.7	8	33	P0.6
Reset	9	32	P0.7
P3.0 Rxd	10	31	\overline{EA} /VPP
P3.1 Txd	11	30	ALE/PROG
P3.2 $\overline{INT0}$	12	29	\overline{PSEN}
P3.3 $\overline{INT1}$	13	28	P2.7
P3.4 T0	14	27	P2.6
P3.5 T1	15	26	P2.5
P3.6 \overline{WR}	16	25	P2.4
P3.7 \overline{RD}	17	24	P2.3
XTAL1	18	23	P2.2
XTAL2	19	22	P2.1
GND	20	21	P2.0

2.5.1.1.Organizace paměti procesoru 89C55

Mikroprocesor má oddělené adresové prostory programu a dat, které jsou přístupné různými instrukcemi. Vnitřní datová paměť RAM 256 bytů je tvořena čtyřmi

bankami (0,1,2,3) po osmi registrech R0,..R7 (adresy 00H – 1FH), za kterými je vyhrazeno 16 bytů pro bitovou oblast. Tyto jednotlivé bity v oblasti jsou přímo adresovatelné. Dalších 128 adres (od 80H až FFH) se využívá k adresování některých významných bitů příslušejících speciálním registrům. Zbývající datová oblast je (30H – FFH) je uživateli volně přístupná. Vnější datová paměť s možnou kapacitou až 64kB je přístupná přes šestnáctibitový ukazatel paměti DPTR, nebo přes osmibitové registry R0, R1 (buď jeden, nebo druhý) [5].

2.5.1.2.Registry speciálních funkcí

Veškeré informace, důležité pro činnost procesoru jsou soustředěny ve SFR - registry speciálních funkcí.

A – střadač, někdy taky akumulátor, je základní registr aritmeticko-logické jednotky. Obsahuje vždy jeden z operandů aritmetické nebo logické operace. Je přístupný nejen běžnými instrukcemi, kde je označen jako A, či pomocí přímé adresy s názvem ACC.

B – registr, jenž je využíván pro instrukce násobení a dělení.

PSW – stavové slovo, obsahuje tyto bity:

C	AC	F0	RS1	RS0	OV	---	P
---	----	----	-----	-----	----	-----	---

C – příznak přenosu

AC – pomocný příznak přenosu, nastavuje se pouze u instrukce DA

F0 – volně použitelný bit

RS1,RS0 – určují registrovou banku

OV – příznak přetečení, nastavuje se při přetečení u aritmetické operace

P – příznak parity, indikuje lichou paritu střadače

SP – Ukazatel zásobníku, je inkrementován a dekrementován při plnění a vyprazdňování. Je umístěn ve vnitřní datové paměti, může být umístěn kamkoliv. Po resetu se nastavuje na adresu 07H.

DPH,DPL – tvoří ukazatele DPTR (DPL – nižší slabiku).

PC – čítač instrukcí, není programově přístupný.

2.5.1.3.Čítače,časovače

Procesor 89C55 obsahuje tři čítače/časovače T0, T1, T2. Jejich obsah je přístupný přes registry TH0,TL0, TH1, TL1, či TH2, TL2. jejich hodinový

synchronizační signál je odvozen od oscilátoru procesoru, nebo z vnějšího zdroje přivedeného na vývody procesoru T0, T1, T2, T2EX. Je-li zdrojem signálu vnitřní oscilátor procesoru, potom čítač ve funkci časovače přičítá jednu jedničku za strojový cyklus, který je tvořen dvanácti periodami oscilátoru. Čítače řídí registr TCON (T0,T1), respektive T2CON (T2).Funkce časovačů a čítačů, i jejich módy jsou podrobně popsány v literatuře [5].

2.5.1.4.Přerušení

Jak bylo již uvedeno, procesor disponuje šesti zdroji přerušení. Vnější zdroje přerušení $\overline{INT0}$ a $\overline{INT1}$ mohou být vyvolána buď logickou úrovní (log. 0), nebo změnou logické úrovně log.1 \rightarrow log. 0. Vznikne-li vnější přerušení, nastaví se příznak IE0, IE1, který se automaticky vynuluje při vyvolání obslužného podprogramu. Přerušení od časovačů 0 a 1 se vyvolávají příznaky TF0 a TF1, které indikují přetečení. Přerušení od časovače T2 se vytváří logickým součtem mezi příznakem TF2 a EXF2 (viz [5]) . Přerušení od sériového kanálu se generuje logickým součtem příznaků RI a TI. Tyto příznaky se automaticky nenulují. Každý se zdrojů přerušení jde povolit v registru speciálních funkcí IE.

IP je registr priority přerušení, v něm se dá nastavit, které přerušení bude mít přednost v případě přerušení v obslužném programu přerušení, či v případě příchodu více přerušení najednou.

Každý zdroj přerušení má svou adresu přerušení, na kterou po přijetí žádosti o přerušení, dokončení rozpracované instrukce a uložení návratové adresy do zásobníku procesor předá řízení. Jejich definice lze vidět v příloze program. Pro návrat z obslužné rutiny se používá instrukce RETI.

2.5.1.5.Sériový kanál

Umožňuje komunikaci ve standardním 8 a 9 bitovém asynchronním režimu nebo 8 bitovém synchronním režimu s pevnou přenosovou rychlostí. Režimy a nastavení se volí v registru SCON a v nevyšším bitu registru PCON. Přenosová rychlost se odvíjí od časovače 1 , či je pevně daná frekvencí oscilátoru procesoru.

Příjmací i vysílací registr je dostupný na stejné adrese, označené jako SBUF. Zápisem do toho registru se naplní vysílací registr, čtením SBUF se přečte hodnota z vyrovnávacího registru, kde je přepsaná hodnota z registru příjmacího.

2.5.1.6.Struktura a činnost vstupně/výstupních bran

Čtyři obousměrné (bitově adresovatelné) brány tvoří záchytné klopné obvody, výstupní budiče a vstupní vyrovnávací paměť. Výstupní vodiče brány 0 a 2 a vstupní vyrovnávací paměť brány 0 jsou použity na styk z externí paměti. V takovém případě je na bráně 0 časově přepínán výstup nižší části adresy pro vnější styk s hodnotou dat, která jsou zapisována nebo čtena z paměti. Z brány dva vystupuje vyšší část adresy. V ostatních případech brána 2 přenáší obsah speciálního vstupně/výstupně registru P2. Brány P1 a P3 jsou uživateli volně k dispozici, s tím, že na bráně 3 a částečně na bráně 1 jsou zavedeny další alternativní funkce.

Chceme-li používat brány P1, P2 a P3 nebo jejich bity jako vstupní, potom v jejich registrech nebo příslušných bitech, které ovládají výstupní tranzistory, musí být zapsané logické jedničky. Teprve potom lze z vodiče číst správnou hodnotu [5].

2.5.1.7.Přístup do vnější paměti

Při spolupráci s vnější pamětí programu se používá řídicí aktivační (řídicí) signál PSEN. Pro spolupráci s vnější pamětí dat se používají řídicí signály RD (čtení) a WR (zápis). Podstatný pro tuto úlohu je styk s externí pamětí dat. Při styku se používá šestnáctibitová adresa (instrukce MOVX @DPTR,A), nebo 8 bitová adresa (MOVX A, @Ri). Při využití celé adresové sběrnice se vyšší byte adresy automaticky vysílá na bránu P2, kde setrvává po celou dobu operace. Nižší byte adresy je vysílán na bránu P0 současně se signálem ALE, který slouží k jeho zápisu do vyrovnávacího registru. Adresový byte je platný při sestupné hraně signálu ALE a proto je vhodné zapisovat nižší byte adresy do registru řízeného logickou úrovní.

Cyklus čtení a zápisu z/do datové paměti je zahájen generováním adresy při signálech ALE, $\overline{\text{PSEN}}$, RD a WR v log. 1. S odstupem alespoň jedné periody hodinového signálu je aktivován signál RD = 0 (WR = 0 pro zápis). Protože instrukce přístupu do vnější datové paměti je dvoucyklová, je aktivní signál RD nebo WR dvakrát delší než aktivní signál $\overline{\text{PSEN}}$. S určitým odstupem před náběžnou hranou signálu RD je pomocí brány P0 procesorem přečtena vybavena hodnota z paměti RAM. S definovaným odstupem po náběžné hraně RD přichází i náběžná hrana signálu ALE, která ukončuje celý cyklus. Aby během celého cyklu mohla být adresa adresovaného místa stabilní, není v první periodě druhého cyklu generován signál ALE [5].

2.5.1.8.Instrukční sada

Vysvětlivky:

A – střadač

Rx – registr R0 až R7

@Rr – nepřímá adresa v R0, R1 (obsah registru určuje adresu místa, s kterým se bude pracovat)

a8,a11,a16 – přímá adresa

#d8, #d16 – data (8, 16 bitů)

r8 – relativní adresa

Přesuny

MOV A,[]

MOV Rx,[]

MOV a8,[]

MOV @Rr,[] - Přesun bytu z paměťového místa na jiné paměťové místo ve vnitřní datové paměti.

Instrukce MOV přesune obsah zdrojového bytu do cílového bez ovlivnění příznaků.

MOV DPTR,#data16 - Instrukce přesune obsah druhého a třetího bytu instrukce do ukazatele dat (DPTR). Druhý byte do DPH a třetí do DPL.

MOVC A,@A+[]

MOVC A,@A+PC - Instrukce přesune byte z programové paměti (operační kód či konstantu) do střadače. Adresa místa,jehož obsah se přesouvá, získáme jako 16-bitový součet obsahu střadače (8 bitů) a ukazatele dat DPTR nebo čítače instrukcí PC. Pro případ čítače instrukcí je jeho obsah před provedením operace inkrementován.

MOVX A,[]

MOVX [],A - Instrukce přesune byte do/z střadače vnější paměti dat. Instrukce mohou využívat 16-bitovou nebo 8-bitovou nepřímou adresu. V prvním případě se vysílá adresa uložená v DPTR na bránu P2 (DPH) a bránu P0 (DPL). Ve druhém případě se na bránu P0 vyšle adresa uložena v registru R0 nebo R1 a na bráně P2 zůstává hodnota naposledy zapsaná.

PUSH a8 - Instrukce přičte jedničku k ukazateli zásobníku a uloží obsah adresovaného místa do vrcholu zásobníku (zásobník je vždy ve vnitřní datové paměti).

POP a8 - Instrukce vyzvedne obsah vrcholu zásobníku a uloží na adresované místo. Pak odečte z ukazatele jedničku.

XCH A, [] - Instrukce výmění obsah akumulátoru s obsahem registru nebo s adresovaným místem.

XCHD A, [] - Výměna nižších „půlbytů“ mezi obsahem nepřímé adresy a akumulátorem.

Aritmetické operace

INC []

INC DPTR - Instrukce přičte k obsahu adresovaného paměťového místa jedničku. Po zvětšení hodnoty FFH dojde k přetečení na hodnotu 00H. Inkrementace obsahu výstupní brány zvětšuje obsah přečtený z registru brány a ne ze vstupně/výstupních vodičů. Instrukce **INC DPTR** přičte jedničku k registrovému páru DPH a DPL, které vytváří 16-bitový ukazatel datové paměti FOTR. Dojde-li k přetečení registru DPL, potom se přičte jednička k DPH.

DEC [] - Instrukce odečte jedničku od obsahu adresovaného paměťového místa. Po zmenšení hodnoty 00h dojde k podtečení na FFH. Dekrementace obsahu výstupní brány snižuje obsah přečtený z registru brány a ne ze vstupně/výstupních vodičů.

ADD A, [] - Instrukce přičte obsah adresovaného bytu ke střadači a výsledek v něm ponechá.

ADDC A, [] - Instrukce přičte obsah adresovaného bytu a příznakový bit přenosu C ke střadači a výsledek v něm ponechá.

DA A - Instrukce koriguje obsah střadače po binárním sčítání dvou dekadických čísel vyjádřených v BCD kódu tak, aby výsledek opět tvořil dvě čtyřbitová BCD čísla.

SUBB A, [] - Instrukce odečte obsah adresovaného bytu od střadače a výsledek v něm ponechá. Je-li požadovaná při výpočtu výpůjčka, nastaví se příznakový bit C.

MUL AB - Instrukce vynásobí dvě osmibitová čísla bez znaménka uložená ve střadači a registru B. Je-li součin větší než hodnota FFH, nastaví se příznakový bit přetečení OV. Jinak se tento bit vynuluje.

DIV AB - Instrukce provádí celočíselné dělení obsahu střadače s obsahem registru B. Celá část podílu zůstává ve střadači, zbytek zůstává v registru B. Při dělení nulou se nastaví příznak OV.

Logické operace

ANL A, []

ANL a8, [] - Instrukce provede logický součin mezi odpovídajícími bity cílového a zdrojového bytu a výsledek uloží do cílového bytu. Je-li cílovým bytem výstupní brána, pak se operace provede mezi výstupním registrem a zdrojovým bytem (nikoliv výstupními signály).

ORL A, []

ORL a8, [] - Jedná se o logický součet, instrukce probíhá podobně jako součin.

XRL A, []

XRL a8, [] - Obdobná instrukce, jedná se o operaci XOR.

CLR A - Instrukce vynuluje obsah střadače

CPL A - Instrukce neguje každý bit střadače a vytváří tak jeho bitový doplněk

Posuny

RL A - Instrukce osmibitové logické rotace akumulátoru o jednu pozici vlevo

RLC A - Instrukce osmibitové rotace obsahu akumulátoru o jednu pozici vlevo přes příznak přenosu

RR A - Obdobná jako **RL**, ale rotuje doprava

RRC A - Obdobná jako **RLC**, ale rovněž doprava

SWAP A - Instrukce prohodí navzájem obsah nižšího a vyššího „půlbytu“ střadače. Je shodná s osmibitovou rotací o čtyři bity vpravo.

Bitové Operace

CLR [] – Instrukce vynuluje adresovaný bit.

SETB [] – Instrukce nastaví přímo adresovaný bit na log. 1.

CPL [] – Instrukce neguje adresovaný bit.

ANL C,[] – Logický součin příznaku přenosu a adresovaného bitu. Výsledek se ukládá do C. Adresovaný bit lze adresovat pouze přímo. Je-li před adresovaným bitem lomítko, součin se provede s negací tohoto bitu.

ORL C, [] – Obdobná instrukce jako výše popsaná, jde o logický součet bitů

MOV C,b8

MOV b8,C – Instrukce přesune hodnotu mezi příznakem přenosu C a daným bitem

Větvení programu

ACALL	a11	- Volání podprogramu uvnitř 2K stránky
LCALL	a16	- Dlouhé volání podprogramu
RET		- Instrukce návratu z podprogramu.
RETI		- Instrukce návratu z obslužného podprogramu přerušení
AJMP	a11	- Krátký nepodmíněný skok na adresu uvnitř 2K stránky
LJMP	a16	- Dlouhý nepodmíněný skok
SJMP	r8	- Krátký nepodmíněný skok na relativní adresu
JMP	@A+DPTR	- Skok na adresu určenou součtem DPTR a obsahem střadače

Skoky

JC **r8**

JNC **r8** – Skok na relativní adresu při nastaveném, nebo nenastaveném C

JB **b8,r8**

JNB **b8,r8** – Skok na relativní adresu při nastaveném či nenastaveném bitu

JBC **b8,r8** – Instrukce testuje adresovaný bit a v případě jeho nastavení (log.1) provede skok na adresu a vynuluje adresovaný bit.

JZ **r8**

JNZ **r8** – Instrukce testuje obsah střadače a v případě jeho nulovosti respektive nenulovosti provede skok na relativní adresu

CJNE A,a8,r8

CJNE[], #d8,r8 - Instrukce porovná střadač nebo registr Rr nebo byte adresovaný registry R0,R1 se zdrojovým bytem (adresa,data). V případě jejich nerovnosti provede skok (maximálně o +127 nebo -127 bytů) na požadovanou adresu. Je-li cílový byte menší než zdrojový, je zároveň nastaven příznak přenosu C=1, jinak je nulován.

DJNZ [],r8 – Instrukce odečte od adresovaného paměťového místa jedničku a zjistí, zda je výsledek nulový. Je-li nenulový, provede skok na definovanou adresu.

Ostatní

NOP – Nic nevykonává, kromě čítače instrukcí nic neovlivňuje.

Pozn: Podrobnější popis procesoru je uveden v příloze ve firemním datasheetu.

2.5.2. Popis displeje

Jelikož výběr displeje nebyl pro zadavatele směrodatný, mělo jít pouze grafický displej, volba padla na levný produkt firmy POWRETIP. Jeho název je PG12864LRU, což značí že má rozlišení 128 x 64 pixelů. Je napájen 5V. Má možnost podsvícení - k tomu slouží vývody označené jako A (anoda) a K (katoda). Dále má možnost volby kontrastu pomocí potenciometru připojeném na vodiči Vo. Data se přenášejí paralelně osmi vodiči (DB0 – DB7). Řídících signálů je pět. E (Enable signal) je potvrzení signálu (na sestupnou hranu se potvrzují data při zápisu, čtení dat je možno, je-li E na log. 1). D/I označuje zda jde o data či instrukci (log. 1 – data, log. 0 – instrukce), a R/W jestli jde o čtení či zápis (log.1 čtení, log.0 zápis). Dalšími dvěma řídícími vstupy jsou CSA a CSB (chip enable). CSA je pro povolení zápisu do levé půlky displeje (sloupec 1 -64), CSB pro pravou půlku (65 -128) [6].

Popis příkazů displeje

Instrukce	R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Display on/off	0	0	0	0	1	1	1	1	1	D

Data z displejové RAM se neobjeví na displeji při D v log. 0. Při změně na log. 1 se na displeji objeví.

Instrukce	R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Display start line	0	0	1	1	Startovací řádek (0 -63)					

Jedná se v podstatě o instrukci posunu displeje v ose Y. Specifikuje řádek v RAM, který se zobrazí na prvním řádku.

Instrukce	R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Set Page (X adress)	0	0	1	0	1	1	1	Stránka 0 -7		

Nastaví stránku (adresu X) RAM ve stránkovém registru.

Instrukce	R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Set adress	Y	0	0	1	Adresa (0- 63)					

Instrukce nastaví příslušný sloupec (Y) , na který se budoucí data pošlou.

Instrukce	R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Status read	1	0	Busy	0	On/Off	Reset	0	0	0	D

Přečte status displeje.

Reset = 1 – systém je inicializován

On/Off = 1 – displej nic nezobrazuje

Busy = 1 – displej vykonává vnitřní operace a je zaneprázdněn. Nepřijímá žádné instrukce.

Instrukce	R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Write display data	0	1	Zapisovaná data							

Zapíše data do displejové RAM.Y adresa se automaticky inkrementuje.

Instrukce	R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Read display data	1	1	Přečtená data							

Přečte data z RAM. Y adresa se automaticky inkrementuje.

Pozn:Podrobné informace o displeji jsou uvedeny v příloze

2.5.3.Paměť EEPROM

Paměť EEPROM na HÚ slouží k zápisu názvu a čísla pokoje dané signalizační jednotky v servisním zadávání. Na každou SIJ v systému připadá textový řetězec o dvaceti znacích, k tomu ještě číslo pokoje na čtyři číslice a adresa. Čtyřciferné číslo pokoje se dá převést na dva byty, textový řetězec musí být ukončen nulou. Počet maximálně možných SIJ v systému je 255. Výpočtem

$$(1+2+21) \cdot 255 = 6120 \text{ bytů}$$

získáme minimální potřebnou kapacitu EEPROM paměti. Takovéto rozsáhlé systémy se však v praxi vyskytují velice zřídka, nejčastější typy provozu mají připojeno do 70

SIJ na jednu HÚ. Tedy $24 \cdot 70 = 1680$ bytů. Vybrána proto byla 16K ($2048 \cdot 8$) paměť, s tím, že do budoucna se tato paměť nahradí pro jistotu větším typem, který pojme všechny informace o teoreticky možných 255 připojených SIJ.

Vybranou paměť typu AT24C16A vyrábí firma Atmel. Jedná se o klasickou sériovou EEPROM paměť v 8 vývodovém pouzdře PDIP. Vývody jsou tyto: Napájení Vcc (5V), uzemnění GND, A_0, A_1, A_2 – jsou adresové vstupy – všechny připojeny na zem. Vývod SCL (Seriál Clock) – vstup hodinového signálu a SDA (Seriál Data) – pro sériová data. Posledním vývodem je WP (Write Protect) – ochrana zápisu. Je-li připojena na nulový potenciál, pak umožňuje čtení a zápis, je-li připojen na Vcc pak chrání paměť proti nežádoucím rušivým pulsům.

Paměť AT24C16 má 128 stránek po 16 bytech a potřebuje na adresaci 11 bitů. Pro komunikaci s nadřazeným zařízením používá specifický protokol. Každá komunikace začíná startovací podmínkou, ta je definována jako sestupná hrana SDA při SCL v log. 1. Končí stop podmínkou, ta je naopak náběžná hrana SDA při SCL v log. 1. EEPROM posílá potvrzení (Acknowledge) v devátém bitu (cyklu) jako log. 0 na SDA. Data na SDA se mohou měnit pouze když je SCL v log. 0.

Při každé komunikaci se rovněž vysílá „device address“ ihned po startovací podmínce. Obsahuje startovací sekvenci na čtyřech nejvýznamnějších bitech tveřenou nulami a jedničkami, dále tři bity které určují adresu stránky a tvoří tak nejvyšší tři bity z jedenáctibitové adresy, a poslední bit určuje, zda jde o čtení (log. 1), či zápis (log. 0).

Tabulka č.4 : Device address

1	0	1	0	P2	P1	P0	R/W
---	---	---	---	----	----	----	-----

EEPROM umožňuje dva druhy zápisu, a to zápis jednoho bytu nebo celé stránky. Zápis jednoho bytu začíná startovací podmínkou, následuje „device address“ s posledním bitem v log. 0, EEPROM reaguje potvrzením, dále se vyše zbylých 8 bitů kam se má zapsat. Opět přijde potvrzení, a může se vyslat 8 datových bitů. Po potvrzení je třeba udělat mikrokontrolérem stop podmínku. Zápis stránky je obdobný, avšak po prvním zapsaném bytu mikrokontrolér nevyšle stop podmínku, ale dalších patnáct bytů. Následuje stop podmínka.

Možnosti čtení dat z EEPROM jsou tři. První z nich je „Current Address Read“ - jedná se o přímé čtení dat z místa, kam ukazoval interní ukazatel paměti EEPROM naposledy inkrementován o jednotku. Po startovací podmínce se pošle jen „device

address“ s posledním bitem nastaveným na log. 1 (čtení), EEPROM reaguje potvrzením a následně vyše data. Řídící obvod poté nevyšle log. 0 na potvrzení dat, ale pouze stop podmínku. Tím je byte přijat.

Sekvenční čtení probíhá podobně, s tím, že po přijetí bajtu řídící obvod reaguje vysláním log. 0, po prvním až předposledním chtěným bytem, a teprve poté následuje stop podmínka.

Naopak pro přímé čtení „Random read“ je třeba vygenerovat první „device address“ s posledním bitem nastaveným pro zápis (log. 0), a po potvrzení vyslat nižších osm bitů adresy. Po potvrzení je třeba generovat další start podmínku, a vyslat stejnou „device address“ s posledním bitem nastaveným pro čtení. Po potvrzení přijdou data. Řídící obvod neposílá log. 0 a vygeneruje stop podmínku [7].

Pozn: Podrobné údaje o této EEPROM jsou uvedeny v příloze

2.5.4.Externí RAM, linkový budič, stabilizátor napětí

Pro zápis dynamických stavů teoretických 255 SIJ, připojených do systému, by datová paměť procesoru nestačila. Proto bylo třeba připojit ještě externí RAM. Byl vybrán typ ze série Hitachi s označením 62256B. Jeho kapacita je 32 768 bytů.

Paměť má 28 vývodů, z nichž je 15 adresových (A0 –A14), 8 vstupně/výstupních na data (I/O0 – I/O 7), dva na řízení čtení/zápisu \overline{OE} , \overline{WE} , a „chip select“ - \overline{CS} . K tomu pochopitelně napájení Vcc a uzemnění GND [8].

Současně s pamětí byl vybrán i registr řízený logickou úrovní, a sice typ 74HCT573. Ten je použit v souladu s procesorem 8055 na zachycení nižšího bajtu adresy (brána P0).

Linkový budič je typu ADM485, je popsán v kapitole 2.2. Všeobecný popis a základní parametry rozhraní RS 485.

Stabilizátor napětí je typ 78L05, stabilizuje z napájení 12V na 5V.

3.Praktická část

V praktické části bude popsáno zapojení a programové vybavení HÚ.

3.1 Zapojení Hlavní ústředny

Schéma zapojení HÚ je uvedeno na obr. č 12. Napájení všech obvodů zajišťuje standardně zapojený stabilizátor.

Procesor Atmel 89C55 je rovněž napájen z tohoto zdroje, vstupy pro oscilátor XTAL1 a XTAL2 jsou připojeny na křemenný oscilátor opracující s frekvencí 11,0592 MHz. Vstup \overline{EA}/VPP je připojen na Vcc. Tím je zajištěno čtení instrukcí z vnitřní paměti procesoru. Všechny brány disponují „pull up“ rezistory o velikosti 22k.

Datový vstup/výstup LCD displeje je zapojen na bránu P0 procesoru. Jeho vývody E, D/I, R/W, CSA a CSB jsou postupně napojeny na výstupy P1.0, P2.0, P2.1, P2.2 a P2.3 procesoru. Napájecí vstupy podsvícení jsou připojeny na zem (K) a na +12V (A).

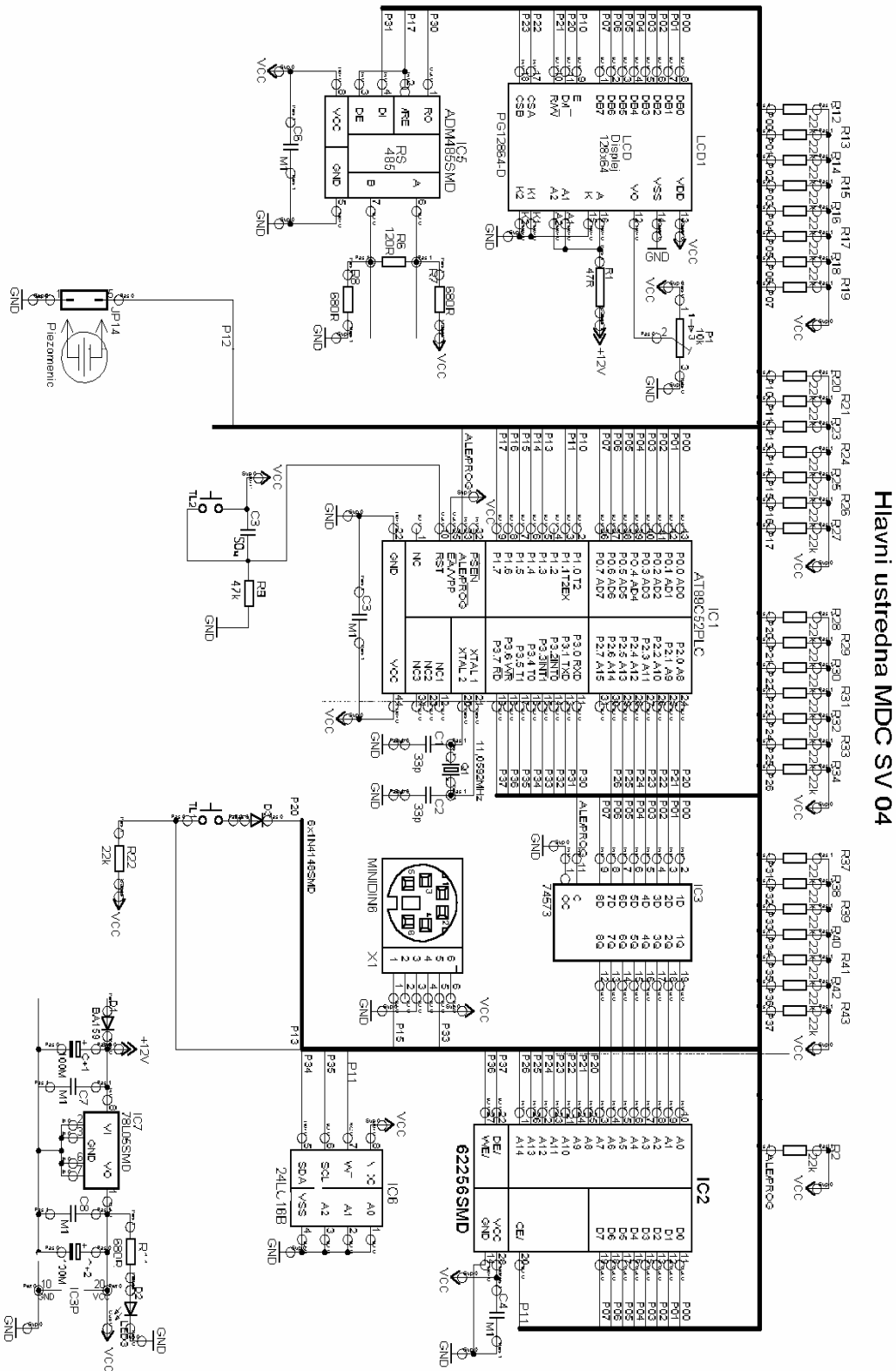
Externí paměť RAM je zapojena v souladu s „návodem“ popsaným v kapitole 2.5.1.7. Přístup do vnější paměti. Celá brána P0 je vyvedena do registru, který je řízen signálem ALE, z něj pak na nižší adresové vstupy paměti. Zbýlých 7 pamětových adresových vstupů je napojeno na bránu P2 (P2.0 – P2.6). Datové vstupy/výstupy paměti jsou opět připojeny na bránu P0. Řídící signály \overline{OE} , \overline{WE} jsou na P3.7 (RD), P3.6 (WR). Signál Chip Enable je připojen na P1.1.

Vývody paměti EEPROM AT24C16 SCLa SDA jsou připojeny na P3.5 (T0) a P3.4 (T1). Write protect je připojen opět na P1.1.

Linkový budič má připojeny vstupy R0 a DE na výstupy P3.0 (RxD) a P3.1 (TxD). Řízení směru toku dat \overline{RE} a DE jsou oba připojeny na P1.7. Linky A a B jsou napojeny na sběrnici RS485.

Tlačítko rušení volání je připojeno přes bránu P1.3 na P2.0, viz schéma. Generátor tónu je připojen na P1.2, a druhý pól je uzemněn. Konektor klávesnice MINIDIN a jeho hodinový výstup je připojen na P3.3 ($\overline{INT1}$), což lze výhodně využít při zpracování dat (na P1.5) z klávesnice přes přerušovací rutinu.

Obr. č.12:Schema HÚ



3.2.Popis programu

Program je celý napsán v assembleru 8051. Má celkově kolem 3500 řádků, obsahuje 62 rutin. Celý je uveden v příloze.

3.2.1.Popis rutin na komunikaci s EEPROM

Protože program aktivně využívá paměť EEPROM připojenou k mikroprocesoru, je třeba se nejdříve stručně seznámit s rutinami zajišťující provoz mezi těmito dvěma periferiemi. Komunikace využívá dva vodiče, SDA – datový a SCL – hodinový. K zajištění komunikace s EEPROM jsou určeny rutiny EERD, EERDB, EEWR, EEWRB. Rutiny respektují protokol popsany v kapitole 2.5.3..

EERD je rutina na přečtení jednoho bajtu z EEPROM. Má dva vstupy – první je ve střadači uložena adresa a „device address“. Načtený byte je uložen do výstupní proměnné B. Rutina v případě chyby v komunikaci s EEPROM nastaví chybové příznaky – CHYBY.0 se nastavuje v případě jakékoliv chyby EEPROM, CHYBY.1 v případě chyby komunikace.

EERDB načítá blok bajtů z EEPROM. Má stejné vstupní proměnné jako EERD, navíc však v B je uveden počet bajtů, které se mají načíst, a v registru R0 je adresa, od které se budou načtená data ukládat do paměti mikroprocesoru. Je vyřešeno i načítání přes více stránek. Jedná se o sekvenční čtení.

EEWR je rutina pro zápis jednoho bajtu se zpětnou kontrolou, která probíhá tak, že zapsaný bajt je zpětně načten a kontrolován, zdali je shodný. Vstupními proměnnými jsou A – adresa, B – zapisovaný byte. Rutina nastavuje příznaky CHYBY.0, CHYBY.1. Příznak CHYBY.2 se nastaví v případě, že se zpětně načtená data neshodují se zapsanými.

EEWRB zapisuje blok byte do EEPROM. Vstupy: A – adresa v EEPROM, odkud se má zapisovat, B – počet byte, které se mají zapsat a v registru R0 adresa v RAM, od které se mají zapisovaná data číst. Lze zapisovat i přes více stránek.

3.2.2.Popis rutin pro LCD

Pro práci s LCD jsou použity dvě základní rutiny LCWRD – posílá data, a LCWRC – posílá instrukce, ze kterých vychází další rutiny. Krom toho existuje rutina LCDINI, která ziniculuje displej.

Rutiny SETX a SETY nastavují sloupec, respektive řádek. Vstupní parametr u SETX je hodnota 0 – 128 v ACC reprezentující sloupec. U SETY jde o hodnotu v ACC 0 – 7, která nastaví stránku řádku.

Rutina LCDCLR vymaže celý displej (posláním 0 na všechny sloupce a stránky řádků).

Na výpis samostatného znaku na displej je použita rutina WRZN. Vstupní hodnota je uložena ve střadači a jde o ASCII kód znaku. Ten se pak konvertuje užitím tabulky nadefinovaných hodnot pro displej. Použitím rutiny LCWRD se znak vypíše na displej.

Poslední používaná rutina je WSTR na výpis řetězce ukončeného 0 z programové paměti. Vstupem je adresa řetězce v DPTR. Rutina využívá WRZN.

3.2.3.Inicializační část

V inicializační části programu jsou nejdříve deklarovány proměnné, vstupy a výstupy. Následuje vymazání celé paměti RAM, nastavení registrů R0 a R1 pro kruhový buffer (na pozdější ukládání a vyhodnocení znaku z klávesnice). Dále se nastaví Stack pointer, režimy časovačů T0,T1 a T2.

Časovač T1 slouží pro sériový kanál, kde přenosová rychlost je 19,4 Kb při frekvenci jednočipu 11,0592 MHz. Časovač T0 slouží ke kontrole stisku tlačítka rušení volání na HÚ, a zároveň zajišťuje kontrolu zvukové signalizace při nouzovém volání. Doba přetečení je nastavena na 2ms. Časovač T2 bude v budoucnu obsluhovat blikání kurzoru.

Následuje povolení a zakázání různých přerušení. V programu se využívají pouze přerušení od časovače 0, kvůli výše zmíněné kontrole tlačítka a řízení zvukové signalizace, dále vnější přerušení 1 – pin INT1, na který je připojen hodinový signál z klávesnice. Toto je nastaveno na sestupnou hranu. Dále se ještě využívá přerušení od časovače T2, na blikání kurzoru na displeji. V této fázi programu je povoleno pouze přerušení od klávesnice. Po vykonání všech instrukcí program ještě počká sekundu. Tato prodleva je nastavena pro možnou připojenou klávesnici. Do této doby se s jistotou rozhodne, zdali klávesnice připojena je, a vyše tedy hlášení o úspěšném diagnostickém testu (AAh), anebo není. Jestliže je tedy zachyceno hlášení od klávesnice v přerušovací rutině, program začne vykonávat podprogram MENU (a

po návratu z něj pokračuje dále). Jestliže ne, program pokračuje do hlavní smyčky. Předtím ale inicializuje displej, vymaže externí RAM, vypíše na displej požadovanou klidovou hlášku s přáním příjemné služby. Nakonec se zakáže vnější přerušení od klávesnice, spustí časovač 0 a povolí se přerušení od něj.

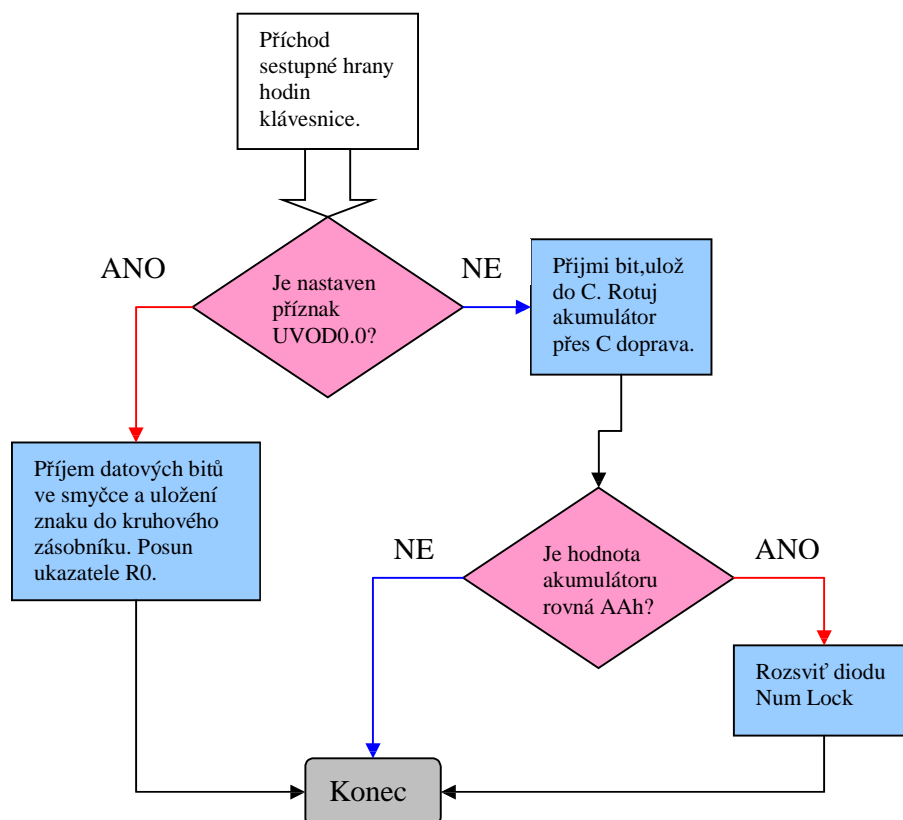
3.2.4.Přerušovací rutina KLAVES od vnějšího přerušení 1

Předtím, než bude popsán podprogram MENU, je třeba seznámit se s obslužným programem pro vnější přerušení. Tato rutina se volá při každé detekci sestupné hrany na vývodu INT1. Je nakreslená na vývojovém diagramu obrázku č.13.

Příznak UVOD0.0 signalizuje, zdali byl přijat signál úspěšného diagnostického testu AAh. Jestliže dosud ne, nasouvají se při každém volání přerušovací rutiny bity do akumulátoru tak dlouho, dokud se nerovnájí právě hodnotě AAh. Když se rovná, nastaví se příznak UVOD0.0. Dále se provede rozsvícení diody NUMLOCK na klávesnici. Nejdříve se stáhne signál DATA od klávesnice na log0, klávesnice pak sám začne generovat hodinový signál. Pošle se tedy příkaz 0Edh, což je úprava stavu diod na klávesnici a k tomu paritní bit (v tomto případě 1). Počká se, až klávesnice pošle potvrzení (0FAh). Poté se vyšle bajt s hodnotou 2h a paritní 0, který zajistí rozsvícení právě diody indikující aktivní numerickou klávesnici. Klávesnice opět reaguje potvrzením.

Při každém dalším vstupu do této obslužné rutiny se již bude jednat o načítání SCAN kódů jednotlivých kláves. Při těchto operacích se již rutina neopouští, dokud nebude načten celý SCAN kód (8 bitů). Ten se pak uloží na příslušnou adresu kruhového bufferu, jako ukazatel slouží registr R0, zálohovaný vždy v proměnné BUFI. Následuje ukončení rutiny a návrat do programu.

Obr. č.13 : Obslužná rutina KLÁVES



3.2.5.Část servisního zadávání

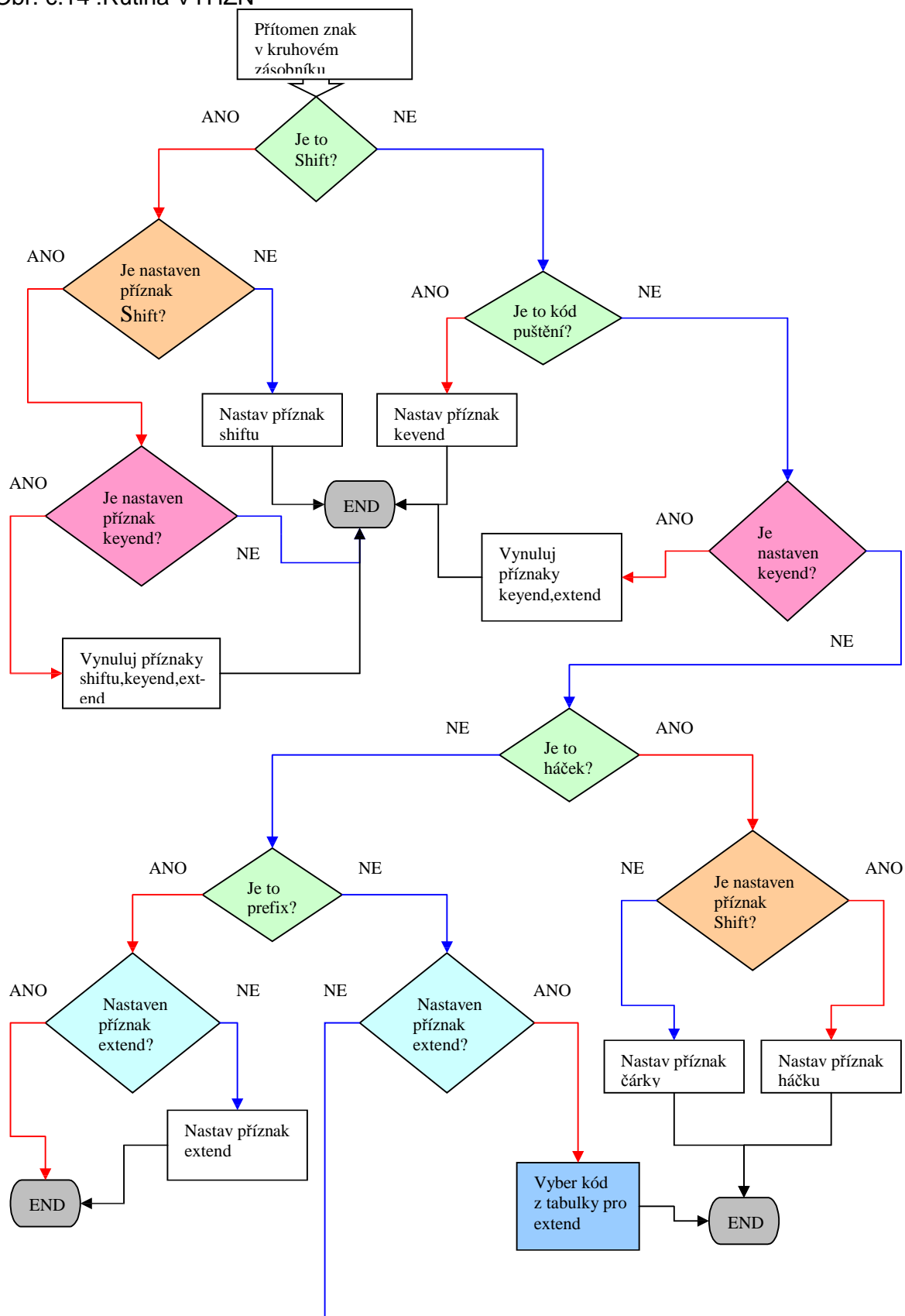
V podprogramu MENU se na obrazovku vypíše dvě položky, které má uživatel možnost zvolit. Je to položka Editační režim, a Sdružený provoz (tato funkce však nebyla předmětem této práce, proto její volba neprovede nic). Dále se v cyklu volá rutina GETKEY. Tato rutina vyhodnotí stav ukazatelů na kruhový buffer, a obsahuje-li R0 větší hodnotu než R1, pak je přítomen znak v kruhovém bufferu. Následuje vyhodnocení znaku.

Zjistí-li rutina GETKEY (požitím rutiny ISKEY) přítomnost nevyhodnoceného znaku v kruhovém zásobníku, zavolá rutinu VYHZN, která vyhodnotí a dekóduje přijatý SCAN kód. Rutina VYHZN je nejlépe popsána obrázkem č. 14.

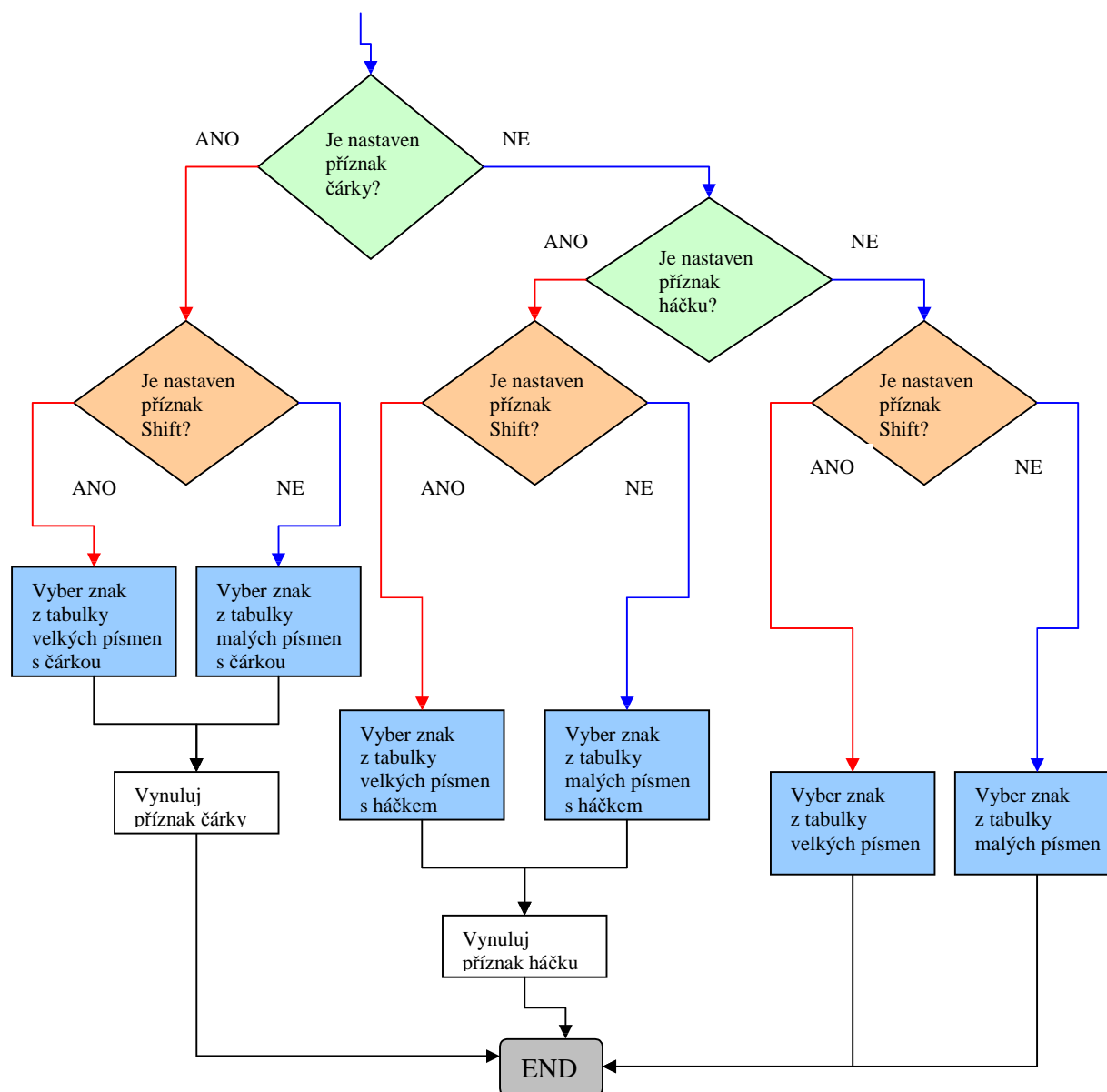
Tabulek znaků je dohromady šest. Jsou uspořádány tak, aby přijatý SCAN – kód odpovídal znaku v ascii, které jsou definovány pro displej. Není-li odpovídající znak v tabulce, je nahrazen nulou, což znamená neplatný znak. Současně je nastaven příznak KEYFLAG0.5, který indikuje, že se program nemá vůbec znakem zabývat. Rutina GETKEY ověří tento příznak, a na základě něj rozhodne, jestli se má znak dále zpracovávat, či nikoli. V každém případě se ukazatel R1 kruhového zásobníku posune o jednu pozici, čímž je vyhodnocení přijatých dat z klávesnice ukončeno.

V podprogramu MENU má uživatel možnost pohybu mezi položkami pomocí šipek nahoru a dolů, potvrzení volby se provádí klávesou enter (kterýmkoliv), a ukončení nabídky se děje klávesou ESC. Po jejím stisku se podprogram ukončí. Stisk jiných než výše uvedených kláves nemá na program vliv.

Obr. č.14 :Rutina VYHZN



Pokračování obr. č. 14: Rutina VYHZN



Vybere-li uživatel položku Editace, program vypíše rutinou PROSTREDI na displej nadpis na první horní řádek, v levém horním rohu na druhém řádku adresa, v levém horním rohu na druhém řádku číslo pokoje a název pokoje ve zbývajících polovině sedmiřádkového displeje. Uživatel, nebo v tomto případě servisní technik provádí editaci tak, že do pole adresa vypíše třímístné číslo adresy (1-256), dále přiřadí k adrese číslo pokoje a jeho název. Na název pokoje je vyhrazeno dvacet

znaků, na číslo pokoje čtyři číslice. Uživatelské prostředí je nastaveno tak, aby odpovídalo použití klávesnice při psaní v programech na PC. Klávesnice je nastavena na českou verzi, klávesy backspace, delete, či posun šipkami vlevo a vpravo pracují tak, jak je obecně zažitá představa. To je v programu obsluhováno rutinami DELETE a BCKSPACE. Přepínání mezi jednotlivými poli je automatické, tzn po vyplnění všech znaků pro pole kurzor přeskočí na pole další. Jinak k pohybu mezi poli jsou určeny šipky nahoru a dolů. To je v programu zajišťováno pomocí rutin NASTAV a NASTAV1.

Příchozí Ascii znaky se při editaci ukládají do RAM paměti procesoru na předem vyhrazených adresách v závislosti, které pole se právě vyplňuje (rutina NOLFOR). Dále je rutinou ZARID zajištěno, aby se do pole ADRESA a ČÍSLO POKOJE daly zapisovat pouze číslice. Při tom se principiálně vychází z hodnoty ASCII, která musí být v rozmezí 30h a 39h.

Po stisku klávesy ENTER či ENTER na numerické klávesnici se provede nejdříve převod adresy z desítkové soustavy na binární (rutina PREVEDA), a převod čtyřciferného čísla pokoje na dvě binární čísla (rutina PREVEDB). Dále se provede zápis do EEPROM. Na to je použita rutina EEWRB. Na získání adresy v EEPROM, na kterou se mají data zapsat, je použit jednoduchý výpočet. Je to adresa SIJ násobeno 24, což je počet vyhrazených bajtů k zápisu (jeden bajt adresy, dva na číslo pokoje, dvacet znaků názvu pokoje a bajt s nulovou hodnotou ukončující řetězec).

Naopak stisk klávesy ESCAPE vyvolá běh rutiny ESC, která vymaže pole název, pokoj a adresa, a způsobí vymazání celého displeje a návrat do hlavní nabídky menu.

Uživatel má dále možnost zjistit, zdali popis SIJ nebyl již předtím zadán a zobrazit si její údaje, a ty případně editovat. Tuto funkci plní rutina SEEK, která se volá po stisku klávesy F1. Stačí tedy vyplnit pole Adresa, stisknou F1 a rutina na základě zadané adresy zkusí najít záznam v EEPROM. Nalezené data se dekodují, tzn. číslo pokoje se převede na desítkovou soustavu a vypíše se na správné pozice na displej. Zároveň se data z EEPROM vyplní paměť RAM určená na pole název pokoje a jeho číslo. Rutina neřeší, zdali byl skutečně záznam dříve vyplněn, anebo ne. V případě dřívějšího nevyplnění se na displej nevypíše nic, neboť data v EEPROM jsou nulové hodnoty, tudíž se jedná o neplatný znak. Podobně se pak do paměti RAM zapíše nuly, což rovněž není na závadu.

Obr č.15: Servisní zadávání



3.3. Komunikační část programu

Jestliže není detekována připojená klávesnice, či je ukončena nabídka MENU, program se začne věnovat komunikaci se SIJ, přičemž musí respektovat komunikační protokol MEDIBUS popsany v 2.3.

Komunikace probíhá s využitím sériového portu procesoru. Sériový port je nastaven do módu 3, což znamená, že pracuje jako devítibitový UART s nastavitelnou přenosovou rychlostí. Ta je nastavena pomocí časovače T1 na 19,4 Kb. V programu není třeba využívat přerušení pro komunikaci.

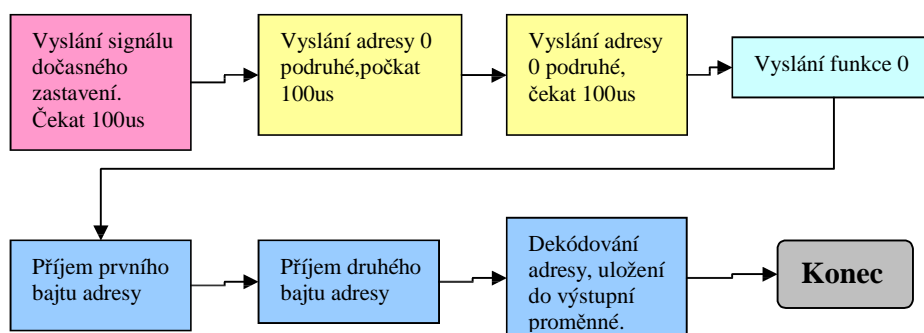
3.3.1. Používané rutiny pro komunikaci

Pro komunikaci se SIJ jsou v programu použity tři rutiny. Ve všech je zakázáno jakékoliv přerušení, z důvodů dodržování časových prodlev. První je rutina na vyčtení dvou adresových bajtů ze SIJ, která vyslala „interrupt impuls“, tedy detekovala změnu svého stavu.

Rutina nepoužívá žádné vstupní parametry, má pouze výstup v proměnné ADRESA. Tato proměnná je vznikne složením dvou adresových bajtů, které se přijmou.

Rutina ABYTE pracuje takto: nejdříve je vyslán signál dočasného zastavení (log 0) o délce 1700us. Ten připraví všechny SIJ na sběrnici na komunikaci. Pak se vyšle adresa s hodnotou 0, přičemž devátý bit TB8 je log 1 (jde o adresu). Následuje prodleva 100us, a adresa se vyšle podruhé. Po další 100us prodlevě se vyšle funkce (TB8 v log 0) s hodnotou 0. SIJ, která vyslala „interrupt impuls“ začne vysílat své dva adresové bajty. Ty jsou vyčteny, dekodovány, uloženy do výstupní proměnné, a tím je rutina skončena. Vyčítání adresových bajtů probíhá v definovaných časových úsecích, kvůli možným chybám v komunikaci.

Obr. č. 16: Rutina ABYTE

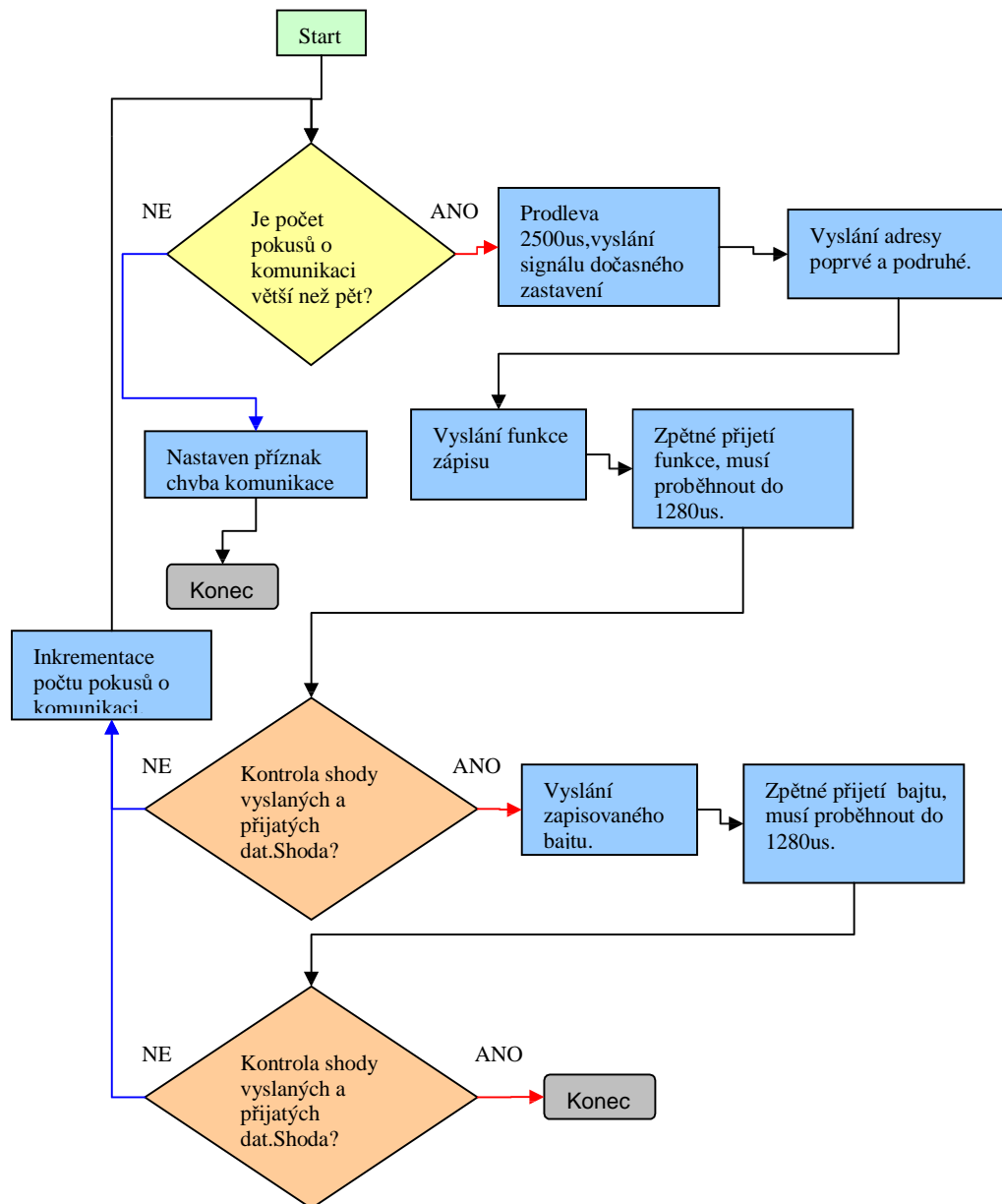


Následující používaná rutina je ZBYTE k zápisu dat na SIJ. Jako vstupní parametry rutina užívá adresu SIJ, na kterou se bude bajt zapisovat, ta je v uložená v proměnné ADRESA, funkci zápisu v proměnné FUNKCE (v našem případě pouze zápis bajtu k obsluze tónového návěští na SIJ) a bajt, který se má zapsat, uložen v proměnné BYTE.

Jelikož je tato rutina volána bezprostředně po sobě v cyklu, kdy se bajt zapisuje na více SIJ, tak se začíná čekací dobou 2500us. To je dvojnásobná doba, po kterou sběrnice musí být v klidu, aby se SIJ připravily na další komunikaci. Dále je vyslán signál dočasného zastavení. Pak se dvakrát vyšle adresa SIJ, následuje vyslání funkce. SIJ pro kontrolu vyšle přijatou funkci zpět. Zkontroluje se shoda. Vyšle se bajt, který se má zapsat, a i ten je načten zpět. Opět se kontroluje shoda. Poté je rutina ukončena. V případě neshody načtené a vyslané funkce/bajtu následuje další pokus o komunikaci, a celý proces se opakuje znova. Je-li

komunikace neúspěšná, tzn. byl překročen počet pěti pokusů o komunikaci se SIJ, rutina nastaví chybový příznak CHYBYK.0 . Rutina je vyobrazena na obr č. 17.

Obr. č. 17: Rutina ZBYTE



Poslední používanou rutinou je CBYTE. Tato rutina vyčte bajt BKOTL ze SIJ. Pracuje podobně jako ZBYTE. Vyšle dvakrát po sobě adresu SIJ, dále funkci čtení dat, zpětně tuto funkci přijme. Porovná je, a v případě neshody se komunikační sekvence opakuje. Jinak SIJ vyšle bajt, který má být načten. Tento se pro kontrolu vyšle ještě jednou. Opět, jestli nejsou oba bajty shodné, celá komunikační sekvence

se opakuje. Jestli nastane shoda, bajt se uloží do výstupní proměnné BYTE, a rutina je ukončena.

3.3.2. Popis komunikační části programu

Po případném ukončení editace anebo v případě nepřipojené klávesnice program zavolá rutinu SIJINI, která má za úkol zinicilizovat SIJ připojené na sběrnici. Inicializace se provádí tak, že se ve smyčce inkrementuje adresa, na kterou se posílá funkce resetu (A0h). Jedná se tedy o využití rutiny ZBYTE. V případě, že komunikace proběhne bez problémů je tedy SIJ na sběrnici vyresetována a připravena k použití. Zároveň se v externí RAM na adresách 0 – 255 začne vytvářet tabulka stavů aktivních SIJ. V případě, že je adresa aktivní, zapíše se na příslušné paměťové místo číslo 2 (vychází se z bytu BKOTL, který se bude později zapisovat na tato místa). V případě opačném se na místo dvojky zapíše 0h.

Poté program po pár grafických úpravách skončí v hlavní smyčce. Zároveň je povoleno přerušení od časovače 0. V hlavní smyčce je testováno, zdali nebyl zaznamenán příchod „interrupt“ impulsu od nějaké SIJ, nebo jestli uplynulo 20 ms, potřebných na testování stisku tlačítka.

3.3.3. Rutina KOMUNIK

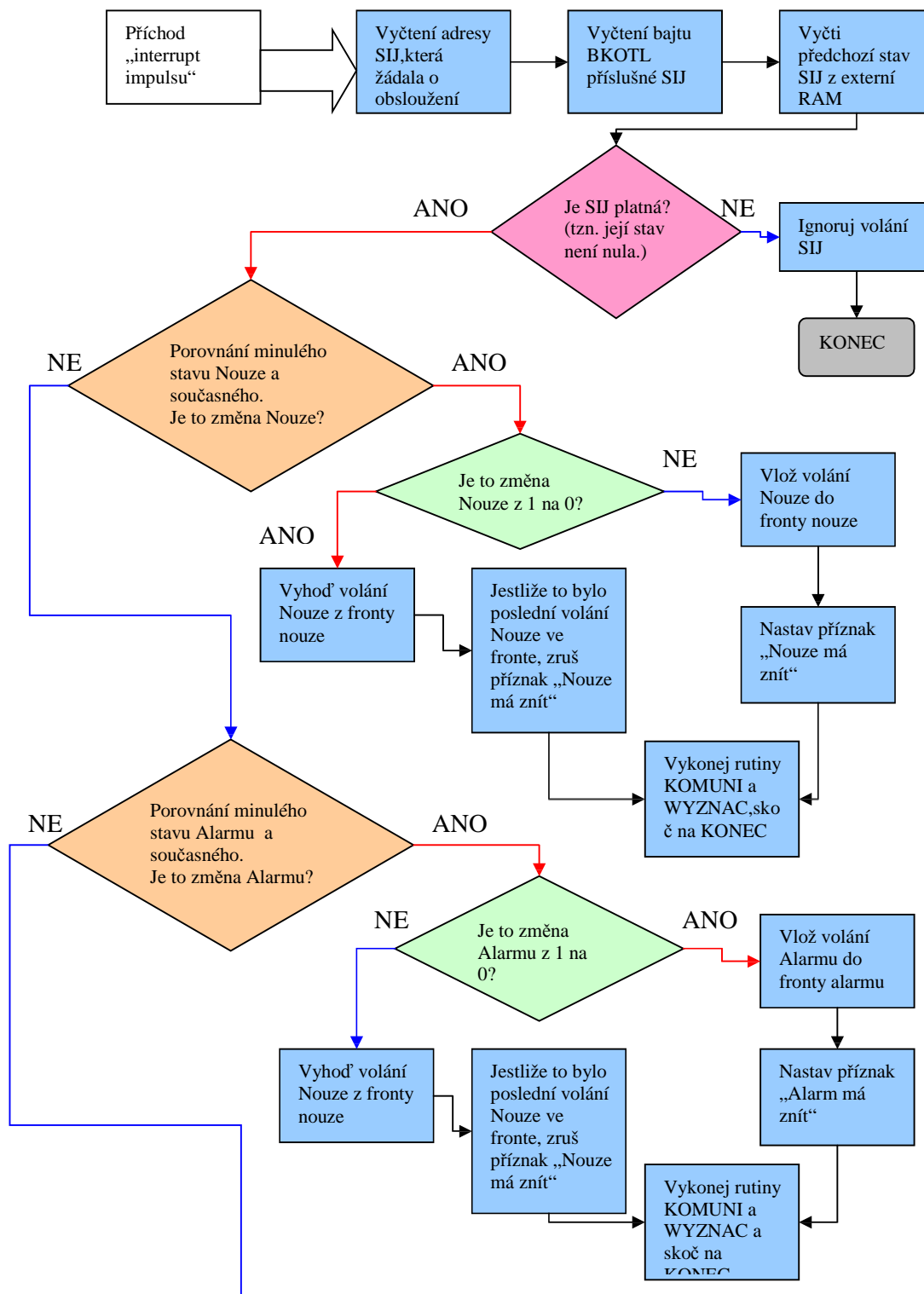
Jestliže je detekován příchod „interrupt impulsu“, zavolá se rutina KOMUNIK. Tato rutina je zásadní pro běh programu. Její princip je vyobrazen na obrázku č.18. Rutina vyčte pomocí podprogramu ABYTE adresu SIJ, na které se změnil stav. Dále vyčte bajt BKOTL ze SIJ pomocí CBYTE. Z externí paměti se vyčte minulý stav bajtu BKOTL na volající SIJ. Následuje porovnávací algoritmus, který detekuje změnu na SIJ. Podle změny, která nastala, se ukládají nebo vyhazují volání do front Nouze, Alarm a fronty registrace Přítomnosti. V případě změn volání nouze a alarmu se rovněž nastavují příznaky „má znít alarm“ a „má znít nouze“, v programu pod názvy FLAG0.4 a FLAG0.2 .

K ukládání volání nouze, alarm a přítomností u do fronty jsou definovány tři proměnné AL, PR, NV. Ty slouží jako počítadla položek ve frontě. Jelikož teoretický počet všech možných volání je 3*255, jsou v externí paměti rezervovány 3 fronty po 255 bajtech. Do jednotlivých bajtů se ukládají adresy SIJ, které volají.

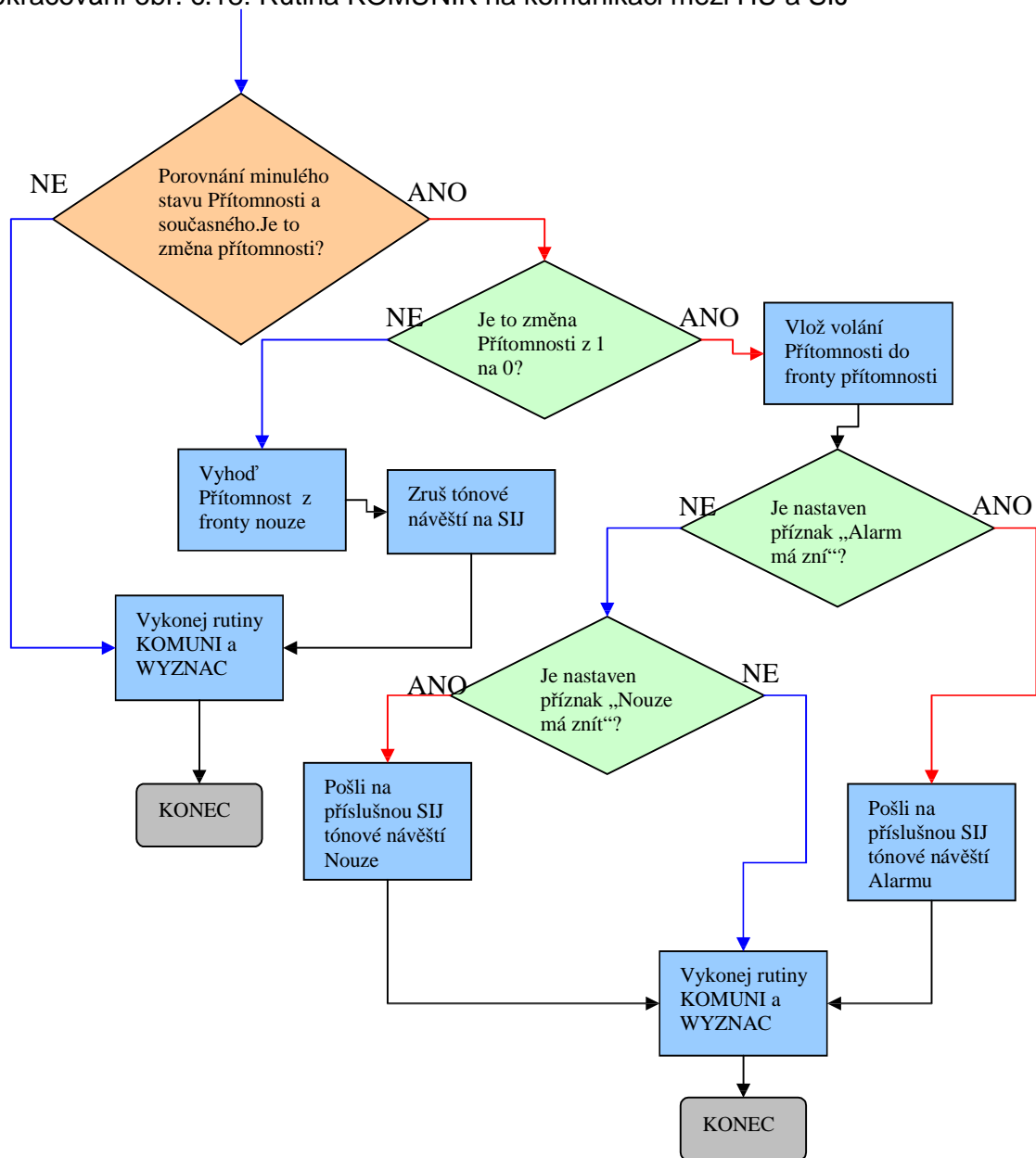
K vyhazování volání a registrací sester z front slouží rutina VYHZFR. Jako vstupní proměnné používá v akumulátoru uložené počítadlo fronty (AL, NV, PR), a paměťovou adresu fronty v DPH. Adresa SIJ, která se má z fronty vyhodit je uložená v proměnné ADRESA.

K obsluze tónových návěstí na SIJ, na kterých je zaregistrovaná přítomnost, se používá rutina KOMUNI. Ta se řídí čtyřmi příznaky. Jsou to FLAG0.5 „zní alarm“, FLAG0.2 „má znít alarm“, FLAG0.6 „zní nouze“ a FLAG0.4 „má znít nouze“. Pomocí jednoduchého algoritmu se na všechny SIJ, na kterých je zaregistrována přítomnost sestry (a tedy jsou uloženy ve frontě přítomností) rozešle příslušný příkaz na rozeznění anebo zrušení patřičného tónového návěstí. Je-li třeba, zapne tónové návěstí alarmu na HÚ, nebo nastaví příznak, FLAG0.1, který určuje, zdali má znít tónové návěstí nouze. Tento příznak se pak vyhodnotí v obslužné rutině pro přerušení od časovače 0. Je-li nastaven, spustí se tónové návěstí nouzového volání na HÚ. Řešeno je to právě takto proto, že zatímco tón pro alarm je nepřerušené pískání (a tedy jde nastavit přímo logickou jedničkou na výstupu P1.2), tón nouzového volání je definován jako rotující maska čísla 15h. Tato maska se dá do střadače, a každé 2,048 ms rotuje doleva přes CY. Tento bit je pak posílán na výstup P1.2, na kterém je připojen generátor akustického tónu.

Obr. č.18: Rutina KOMUNIK na komunikaci mezi HÚ a SIJ



Pokračování obr. č.18: Rutina KOMUNIK na komunikaci mezi HÚ a SIJ



3.3.4. Rutina WYZNAC

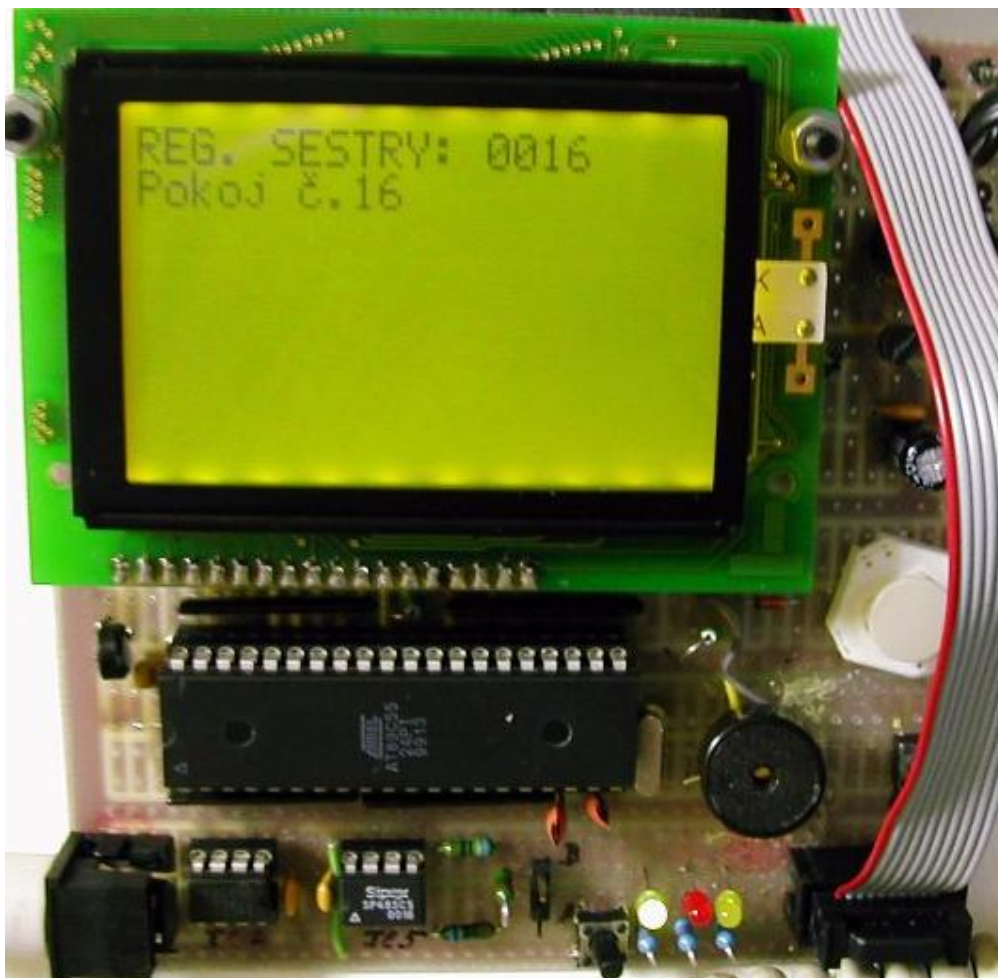
Poté co byla obsloužena tónová návěští je třeba provést záznam o volání na displeji HÚ. Na LCD je místo pro tři záznamy o volání (vždy dva řádky na volání, jeden řádek na hlavičku a poslední řádek je volný na chybová hlášení). V této fázi se z KOMUNIKu zavolá podprogram WYZNAC. Tato rutina nejdříve zjistí stav fronty na alarmy. Není-li prázdná vypíše první tři alarmy ve frontě. Je-li počet alarmů menší jak tři, doplní zbývajících místo na výpis nouzovým voláním. Zrovna tak, je-li fronta vyhrazená alarmům prázdná, vypíše se pouze nouzové volání. Neexistují-li v systému právě volání alarmu a nouze, vypisují se na displej přítomnosti sester z fronty přítomností. Jsou-li všechny tři fronty prázdné, není tedy co vypisovat, a na displeji se objeví přání příjemné služby.

Z důvodu větší přehlednosti programu rutina WYZNAC volá vlastní subrutinu WYZ. Ta zajišťuje výpis údajů o SIJ na displej. Jejími vstupními parametry jsou adresa SIJ v proměnné TEMP a v proměnné HELP je uložen typ volání (0 – alarm, 1 – nouzové volání, 2 – přítomnost). Podle adresy se v EEPROM najde existující záznam – číslo pokoje a název – o SIJ, který se vypíše na displej.

Obr. č.19: Volání alarmu na pokoji 20 „rehabilitace“ a NV na pokoji č 16.



Obr. č. 20: registrace sestry na pokoji č. 16



3.3.5. Funkce tlačítka na HÚ

Tlačítko na HÚ má za úkol při svém stisku zrušit první volání v řadě. Ruší pouze alarmy a nouzové volání.

Stav tlačítka se testuje vždy po 20 milisekundách. Tuto dobu odměřuje časovač 0. Ten má nastavené přerušení po 2.048 ms, tedy vždy po deseti přerušeních nastaví příznak FLAG0.0. V hlavní smyčce se při jeho logické 1 provede skok na návěští TLAC. Zde je stav tlačítka vyhodnocen, a je-li zaznamenán stisk a puštění, zavolá se rutina ZRUSVOL.

Tato rutinka nejdříve vyhodnotí stav fronty na alarmy. Nachází-li se nějaký alarm ve frontě, zavolá se dvakrát rutina VYHZFR, a provede se odstranění prvního alarmu a zároveň odpovídající přítomnosti z front. Je-li fronta alarmů prázdná, testuje se fronta na nouzové volání. Po případném odstranění nouzového volání se ještě testuje, zdali nebyla současně zaregistrována na SIJ přítomnost. Ta by se rovněž

odstranila z fronty. Jsou-li fronty alarmů a nouzových volání prázdné, neprovede se nic.

Po odstranění volání z front se volá rutina WYZNAC. Tím se obnoví aktuální stav systému na displeji. Dále se zavolá rutina KOMUNI, přičemž se rovněž obnoví správný stav tónových návěští jak na SIJ tak i na HÚ.

Nakonec se provede reset SIJ, jejíž volání bylo zrušeno. Na to se použije rutina ZBYTE, se stejnou funkcí (A0h) a zapisovaným bajtem (A0h) jako u inicializace SIJ. Zde je nutno podotknout, že toto řešení je pouze provizorní. Funkce rušení volání totiž u SIJ zatím neexistuje.

3.3.6.Ošetření chyb

K chybě při komunikaci mezi SIJ a HÚ může dojít, pokud SIJ nedodrží časové rozvržení komunikace, odpovídá nesprávně, či nekomunikuje vůbec. Jestliže se tak stane, projeví se tento fakt v rutinách ZBYTE a CBYTE. Ty nastaví chybový příznak CHYBYK.0. Po každém návratu z těchto rutin se tento příznak testuje, a jestli nabyde hodnoty logické 1, zavolá se rutina VYPIS.

Ta zjistí stav SIJ z externí paměti a její volání vyhodí z front. Dále do tabulky stavů v externí paměti zapíše 0. Tím je zajištěno to, že v rutině KOMUNIK budou veškeré volání z této SIJ ignorována a neovlivní chod systému. Na poslední, sedmý, řádek displeje se vypíše chybové hlášení o vyřazení inkriminované SIJ, avšak kvůli nedostatku místa se nevypíše celý název pokoje ani číslo, pouze adresa.

Po návratu z rutiny VYPIS se vždy volají rutiny WYZNAC a KOMUNI. Systém se tedy vykazuje dobrou stabilitu, dokáže se zotavit z kolapsu SIJ. Dále komunikace probíhá bez problémů.

4.Závěr

Cílem této práce bylo realizovat návrh hardware a software řídicí jednotky systému MDCS. Všechny body této práce se podařilo splnit. Autor se seznámil s vlastnostmi a programováním mikroprocesorů řady 8051, s komunikačním protokolem MEDIBUS. Navržené obvody hlavní ústředny splňují všechny požadavky. Připojení standardní AT klávesnice se podařilo rovněž zrealizovat. Při testování funkčního vzorku hlavní ústředny v systému MDCS s více zapojenými signalizačními jednotkami se program projevil stabilní.

Aby se stávající funkční vzorek mohl stát plnohodnotnou hlavní ústřednou systému MDCS, musí se ještě provést řada vylepšení. Do zapojení bude přidán watchdog, přidají se telefonní linky, přidá se možnost poslouchat zábavný program. Implementace protokolu MEDIBUS do hlavní ústředny bude úplná. Rovněž se musí přidat světelná signalizace.

5. Seznam literatury

- [1] Komunikační jednotka – firemní literatura ZPT
- [2] Protokol MEDIBUS – firemní literatura ZPT
- [3] Kolektiv autorů, PC v tabulkách, 2. přepracované vydání, 1997 UNIS Publishing s.r.o.
- [4] Interfacing PC keyboard [online] [cit. 4.11.2002] dostupné na internetu
<http://www.beyondlogic.org/keyboard/keybrd.htm>
- [5] Petr Skalický, Mikroprocesory řady 8051, Praha, nakladatelství BEN 2000
- [6] Katalogové listy pro displej PG12864, firma POWER TIP
- [7] Katalogové listy pro EEPROM AT24XXX, firma ATMEL
- [8] Katalogové listy pro RAM 62256, firma Hitachi